User Guide: Map Translator Control

User Guide: Map Translator Control

Table of Contents

Preface ix
1. Introduction 1
    What Is the Map Translator? 1
    Why Translate Maps? 1
    What Is Involved? 2
2. Overview 3
    System Requirements 3
    User Requirements 3
    Terminology 3
        Name 3
        Label 4
        Data Source 4
        Data Type 4
        Field 4
        Style 4
    Standard Controls 5
        Tabs 5
        Map Navigation Buttons 6
        Command Buttons 6
            Applying Changes 7
            Deleting Items 7
            Saving Changes 8
        Information at a Glance 8
        Entering Information 9
    Configuration Windows 9
        Saving Configuration 9
        Exiting 9
        Help 9
3. Getting Started 10
    Goal 10
    Step 1: Install the Software 10
    Step 2: Launch the Map Translator Control 10
    Step 3: Configure Your Mapping System 10
        Specify Data Sources Configuration Tool 10
            Configure SDE Database Parameters 11
            Configure Personal Geodatabase Parameters 12
            Configure AutoCAD Parameters 12
            Configure Shapefile Parameters 13
            Configure GE Smallworld Parameters 13
            Configure TIGER Files Parameters 13
    Step 4: Adding Other Data Sources to Maps 14
    Step 5: First Translation 15
    Step 6: Verify Results 15
    Step 7: Proceed with Configuration 15
4. Translator Configuration 17
    Introduction 17
    Setting Background Color and Centering Point 18
    Controlling Scales and Visibility 19
        LOD Contents Tab 20
        Configuring Scales 20
        Configuring Visibility 21
            Configuring the Key Map 22
            Configuring Navigation Zoom Levels 22

Configuring the Finer Zoom Levels 23
Configuring Lines, Points, and Text 23
Configuring Data Types 27
Getting Started 27
Configuring Data Type Names 28
Setting Data Type Labels 29
Enabling and Sorting Data Properties 29
Setting Map Viewer Labels 29
Saving Your Work 29
Labelling 29
Getting Started 29
Creating a New Label 29
Naming Labels 30
Selecting What to Label 30
Configuring Placement and Location of Labels 30
Configuring Label Text Settings 31
Line Labelling 32
Adding the Label to Translation 33
Saving Your Work 33
Labelling Examples 33
Placing the KVA inside a Transformer Symbol 33
Labelling Primary Line with Conductor and Phase 34
Labelling Primary Line with Flow Direction Arrows 34
Configuring Find Items 35
Getting Started 35
Creating a New Find Item 35
Setting Find Item Conditions 35
Selecting the Sources 36
Saving Your Work 36
Removing Find Items 36
Configuring Block Explosions 36
Merging External Data 37
Creating a New Data Extraction (Dump File) 38
SQL Only: Specifying How to Read the Data Source 39
Adding a Merge Transform for the Data Source 40
Configuring the Map Translator to Merge the Data 41
Configuring Python Transforms 41
Troubleshooting 43
Configuring Propagation 43
Configuring Partner's Palm Map Translator 44
Requirements 44
Configuring the Palm Translator 44
5. Maintenance & Production 45
Daily Routine 45
Step 1: Translate Maps 45
Step 2: View Maps 45
Step 3: Post Maps to the Network 45
Reconfiguration 45
Installing and Updating Map Viewers 45
Installing the Map Viewer from a CD 45
Installing the Map Viewer from the Network 46
Updating the Map Viewer 46
Running the Map Viewer Directly from the Network 46
Automatic Translation: Configuring Windows Task Scheduler to Translate Maps 46
Automated Email Notification 48
Components 48
Configuration Details 48
Troubleshooting 49

Reopen All Drawings 49
Zip Up Logs 49
Zip Up Configuration 49
Interpreting the Logs 49
Translation Fails with a UTFDataFormatException 49
Translation Fails with OutOfMemoryException 49
Bad Command or Filename Message when Running a Batch File 49
6. Advanced Configuration 51
Data Type and Style Merging and Splitting 51
Configuring Nomenclatures 51
More Translation Options: MapTranslator.tof 55
Ignoring Tables in ESRI 56
ESRI Rotation Fields 56
Adding Pictures and Other Raster Images 57
Configuration 58
Adding Construction Job Information 59
Symbols 60
Creating a Symbol File 60
Special Characteristics 60
Management of Redline Symbols 61
Management of Mapping Symbols 61
Management of Company Logo 61
Changing the Color of Your Find Item Arrow 61
7. Importing ESRI Shapefiles from the Partner Field Designer into AutoCAD 62
Components 62
Turning on Shapefile Export in the Field Designer 62
Creating the Data Items for Export 62
Importing Shapefiles into AutoCAD 64
Configuring the AutoLSP Import Program 65
Creating a Saved Loading Configuration 67
Loading the LSP Program into AutoCAD 67
Running the LSP Routine 67
8. Python Scripts 68
How Python and Partner Interact 68
Using Python 70
String Manipulations 70
Error Handling: Try—Except 71
..... 71
Examples 72
Formatting a Telephone Number 72
Splitting Graphical Elements 72
Regular Expression 73
Converting Multiple Merge Fields to Individual Data Fields 73
Counting the Number of Objects in a Multiple Merge 74
Building a Python Transform 74
A. Colors 75
B. XMLGenerator—A Data Dumper for GE Smallworld 76
Using the XMLGenerator GUI 76
Using XMLGenerator at the Magik Prompt 78
Index 82

List of Figures

2.1. Partner's Map Viewer Opening Screen 5
3.1. Specify Data Sources Opening Screen 11
4.1. Map Translator Control Panel 18
4.2. General Map Settings 19
4.3. Configuring the Translator 20
4.4. Configuring Scales 21
4.5. Configuring Visibility 22
4.6. Configuring the Map Viewer 24
4.7. Configuring Linetypes 25
4.8. Configuring Text 26
4.9. Configuring Data Types 28
4.10. Adding a New Label 30
4.11. Configuring Placement and Location of Label 31
4.12. Configuring Text Labels 32
4.13. Configuring Line Labelling 33
4.14. Configuring Find Items 35
4.15. Configuring Block Explosions 37
4.16. Configuring Other Data Sources 38
4.17. Configuring Merge Transforms 40
4.18. Configuring Python Transforms 42
4.19. Configuring Propagation 43
6.1. The Map Viewer with highlighted padmount transformer and button added for linked picture. 57
6.2. Padmount transformer picture. 58
B.1. XMLGenerator GUI 77

List of Examples

6.1. A Nomenclature in Maps/resources/MapControl.tof 53
6.2. Multiple, Independent Nomenclatures example 55
6.3. 56
6.4. Sample external CSV file linking map location and photo name. 58
6.5. Editing the MapDataSource file. 59
6.6. Python transform text. 59
7.1. Creating a CSV File in a Spreadsheet 63
7.2. Creating a CSV file in a Text Editing Program 63
7.3. Setting for Field Designer to Export a Shapefile Called Unitpoints 64
7.4. Listing of ShapeFileImport.lsp 65
B.1. Visual Basic 6.0 Application Using the CLI Commands 79

Preface

Apple Computers aired a commercial a few years ago that did an excellent job of skewering the state-of-the-art in personal computing. In it, a father buys his son some sort of CD-ROM game involving dinosaurs. They bring it home to install, but, of course, drivers and software have to be updated, the case has to come off, wires have to be checked and moved, etc. All through this process the child pleads, "I wanna see the dinosaurs." Finally, with the computer in pieces all across the table and the father exhausted, the child takes the CD and heads out the door. "I'm going to Jimmy's house. He has a Mac."

At least, it went something like that.

This is the manual for the most complex piece of software we have ever written. It is the culmination of some seven years of effort in making map viewing and publishing fast, efficient, and easy. Recently, we have even made great strides in making the actual configuration, maintenance, and translation of map sets easier. There is still some work to be done, but we hope that this manual, with a little patience and determination—but not too much—on your part, will allow you to configure, distribute, and maintain a map viewing system throughout your organization, finally bringing to light the hard work and hard cash that have gone into building your mapping and other information systems.

If you want to understand the software, its purpose, the requirements, and proceed step-by-step through the process of configuration, translation, and distribution, please proceed more or less straight through the manual as written.

If, however, you just want to see the dinosaurs, print out the *Quick Start* included with the software or skip to *Getting Started*.

Chapter 1. Introduction

What Is the Map Translator?

All of Partner Software's applications have bland-yet-descriptive names. We've tried to avoid being cute. Unfortunately, some of the names can be confusing due to their generality.

There are many tools which translate maps or GIS data from one format to another. The Partner Map Translator is not such a a general-purpose tool, however. It translates maps, GIS data, and a variety of other associated data sources into our own proprietary format, which is optimized for read-only map viewing.

The configuration of this translation process, which is the primary topic of this manual, allows the careful merging and tailoring of the map data to meet the end user's (those using the map viewer) needs. This ranges from simply changing the name of a data field to something more intelligible, to complex combinations and transformations of the data itself. An important part of this process is the elimination of extraneous data—data that might be critical to the mapping or GIS system but which is of no use to users viewing maps.

Although not technically part of the translation process, another task addressed by the software is that of defining the legends—the colors, icons, and text styles used by the Map Viewer.

And finally, the Map Translator software includes our propagation tools, which allow automated updates of all installations of the Map Viewer, including those at remote offices.

Since the resulting maps are read-only, it is important to remember that the Map Translator and Map Viewer are not a mapping system or GIS themselves. One way we like to think of them is as an interactive map publishing system—instead of producing expensive paper maps, you are producing continuously-updated electronic maps for much less cost and effort.

Why Translate Maps?

Those initially exposed to our technology often wonder why we go through this process at all. After all, most GIS and CAD systems have some sort of viewing software, or you can simply use the GIS or CAD system directly. And there are viewers which can directly read GIS and CAD formats.

There are quite a few reasons behind our approach—based on useability, technical, and business concerns.

As far as ease-of-use goes, we decided early on that standard GIS and CAD systems, and often their associated viewers, are far too complex to use and maintain. Generally, this software is designed for mapping personnel rather than nontechnical users.

Some systems, especially those based on CAD software, have built-in limitations that conflict with user needs. For example, many AutoCAD-based map systems consist of many separate drawings, rather than a single seamless map.

This reveals some of the technical problems. It is not unusual for an organization to have more than one source of map data—perhaps a land base in one format and an electrical grid in another. Also, commonly there is associated information—customer information, equipment details, etc.—that could be associated with the maps, but are not due to the high cost of importing and maintaining the data in the GIS. Presenting a unified interface to these different systems is very complex and tends to be fragile.

There are numerous problems with speed and scalability. GIS systems are designed to be both viewed and maintained. This limits their speed in various ways, since data consistency must be ensured. Also,

they tend to scale poorly—the larger the map set, the slower they get, or you have to make concessions such as splitting up the map set into separately-maintained chunks. Our format, on the other hand, is designed exclusively for viewing and query speed. It scales extremely well, allowing map sets of virtually any size and complexity.

Another facet often neglected in other map viewing systems is the problem of updating installations. Not all computers are connected to the network continuously, and updating their copy of the maps can be very time consuming. Most mapping systems require copying a complete set each time an update is needed. This can be very slow when the maps are hundreds of megabytes in size, and eventually causes the maps to be updated infrequently. Our map format, however, was designed with updating in mind and minimizes the number of changed files to an amount that reflects the actual updates to the map's data. Our propagation system also utilizes on-the-fly compression to speed transfers.

Finally, there is the very pragmatic concern of tying our pricing or technology to any one vendor. There are toolkits available from all major mapping vendors that allow embedding map views in software, but all of them restrict the types of information you can display, and virtually all of them also have an associated per-seat royalty. This is unacceptable for us since we sell our software exclusively with site-wide licenses. You pay for the core software and maintenance but can install as many viewers as you like without need of tracking licenses or paying for each user.

What Is Involved?

Using the Map Translator consists of a moderate amount of configuration and setup effort, followed by a simple routine for maintenance and updates.

You can generally get your maps into the Map Viewer quickly—15 minutes to an hour. Our *Quick Start* guide and *Chapter 3: Getting Started* are designed to get you over this first hump. Most people find that after that they spend a fair amount of time polishing the map set—changing labels, colors, icons, etc. to get just the right look. You can do this all at once or continually improve it over time as you get feedback from the users.

Once the map set is configured to your liking, maintenance is straightforward. Whenever you have made significant changes to the GIS or on a regular schedule, run the translation, verify that it completed correctly, then post the results to the network for use by the Map Viewer. Most of our customers do this once a day or so. Depending on the size and complexity of the map set, translation can take from 5 minutes to an hour—so it can certainly be done over a lunch break.

If you make changes to the structure of your GIS or map system or you want to add additional data from other sources, you will need to do some reconfiguration. As long as the data sources are correctly identified, the software is designed to be aware of structural changes, so generally reconfiguration is just a matter of adding or removing features from the map sets, naming fields, updating the legend, etc.

Occasionally, Partner will release updates for the software itself. Installing the updates and propagating them to the Map Viewer installations are very straightforward.

Chapter 2. Overview

The Map Translator application consists of a suite of configuration and production tools, all run from an initial window called the *Control Panel*. Almost all the configuration is controlled by graphical tools; some portion, however, does still require editing configuration files by hand. We have tried to make the tools as consistent as possible to minimize user confusion; we'll discuss the standard types of windows and controls in this chapter.

System Requirements

The following requirements for the Map Translator computer vary depending on the size and nature of the map data sources. Translation of AutoCAD-based mapping systems requires at least 256 MB RAM.

- Greater than 300 MHz processor.
- At least 128 MB RAM—the more RAM the better.
- 2 GB hard disk space available. For AutoCAD map systems, the hard disk should have space available that is at least 4 times the size of the drawing set.
- Windows 95, 98, ME, NT, 2000, or XP operating systems.
- TCP/IP compatible network, such as Windows NT and Novell.
- 800 × 600 screen resolution.

User Requirements

You should be able to download, view, and execute files in Microsoft Windows and that you are familiar with your GIS or mapping system. Do your best to get through this guide, but feel free to call us at 800-964-1833 if you get stumped.

Since the Map Translator Control module is configured for your map set, which can be based on shape files, drawing files, or database files, only use the parts specified for your configuration.

Terminology

We have tried to be consistent in the use of some standard terms throughout this manual and the software.

Name

Almost everything in the Map Translator has some sort of name—a word or words used to refer to the thing in lists and other controls. Examples of things with names include visual styles, data types, data fields, GIS files, data sources, etc. Sometimes names come directly from the source data read by the translator—for example, the name of an AutoCAD block or an ESRI table. Sometimes names are derived using a naming scheme from other data—for example, styles and data types are named according to GIS format-specific rules. Sometimes names are set by the user—for example, when you name an external data source or set the style name for a label.

One general rule applies throughout the software—when configuring the Map Translator, you always

refer to things based on their names, not based on their labels (see the definition for label below). Keep this in mind when configuring find items and other features that require reference to other things in the Map Translator.

Label

Label, on the other hand, generally refers to the text label used to refer to things in the final product—the Map Viewer. Examples include field labels and data type labels. Some things, such as find item types, use their name as their label, and therefore do not have a separate label.

Unfortunately, label has another meaning, to refer to text or point labels generated on the map itself. This type of label is always referred to as a *Map Label*, and its usage should be clear from context.

Data Source

A data source is a set of data read by the Map Translator. Data sources include GIS data and external data merged with the GIS data. Examples include AutoCAD drawing sets, ESRI geodatabases, and SQL queries.

Data Type

A data type defines a category of similar data that has the same data fields. Data types are originally established when reading the GIS data, and the available fields are determined by examining the incoming data. Data types are referred to by a data type name or, in the Map Viewer, by a label. This label appears at the top of the *View Data* listing in the Map Viewer, to help the user know what kind of data item they are viewing.

Field

A field defines an individual data value for a data type. Fields can contain dates, numbers, or text strings. Fields are referred to by name and also have an associated label in the Map Viewer. Fields can be disabled in the Map Translator. This means they are not translated and are not available in the Map Viewer; however, they are available during translation for tasks such as map labelling, data merging, or other uses. Data fields are similar to variables, columns, or properties as used databases, spreadsheets, and other software.

Style

A style defines a category of graphical data that can be viewed on the map itself. Styles are broken into three distinct categories: line styles, point styles, and text styles. Styles are referred to by name, and the name is always prefixed with the type of style: "line-", "point-", or "text-".

The difference between styles and data types can be confusing. Consider the following distinctions:

- styles refer to drawable geometry data;
- data types refer to the data shown in the *View Data* tab;
- a graphical entity from a GIS system generally consists of one data item and one graphic; after translation these may result in one data item with several associated graphics of different types;
- style names always have a type prefix: "line-", "point-", or "text-";
- find items are based on data types, not styles;

- visibility is based on styles, not data types.

In general, you will have many more styles than data types. This is because, while a set of line or point data may share the same data fields, you may wish to set them to different colors in the Map Viewer.

One excellent example of the difference between styles and data types is AutoCAD block explosion—here an original point object (a block insert) can be exploded into its component lines and text attributes, generating many different graphical styles, including the original point. However, there is still only one insert, and therefore only one data type.

Standard Controls

Each Partner Map Viewer has an opening screen similar to the one below with a *Key Map* on left side and the *Control Panel* on the right.

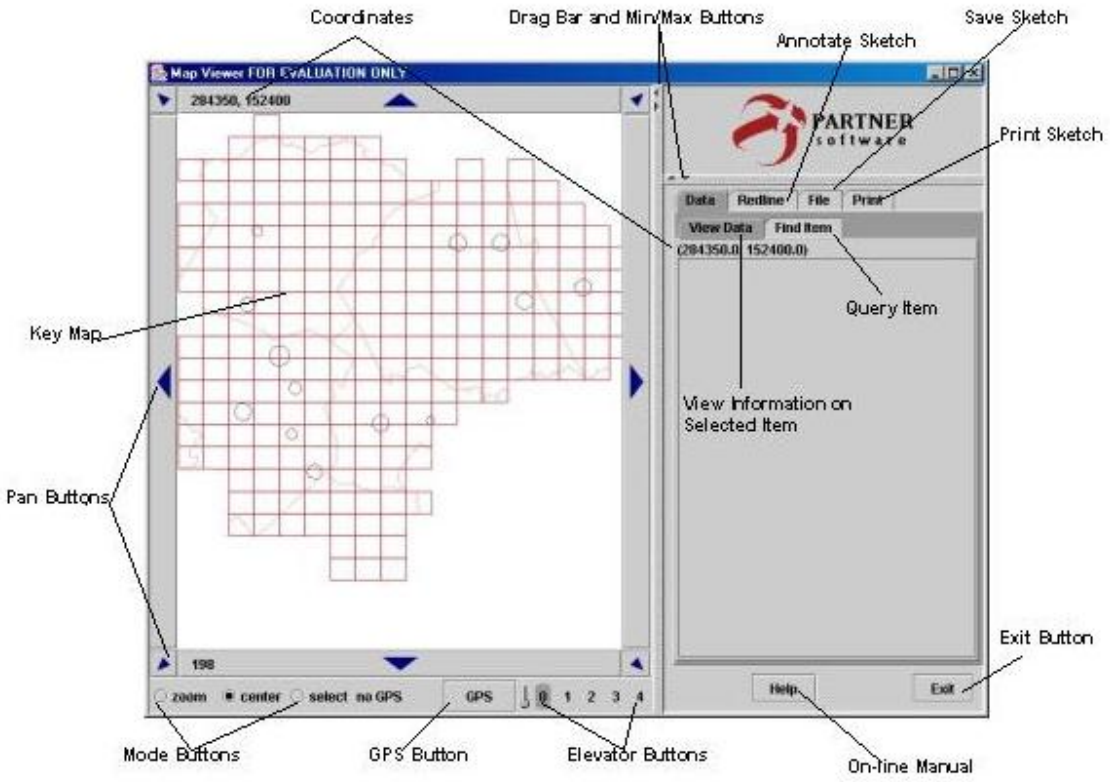


Figure 2.1. Partner's Map Viewer Opening Screen

Tabs

On your opening screen, the *Control Panel* contains a logo (either Partner's logo or your logo) and the *tabs* to control the information displayed on your screen.

Tabs control the information displayed on your screen.

- **Primary Tabs**: Tabs located at the top of your screen, which control the information you see.

- **Secondary Tabs**: Tabs located under one of the primary tabs at the top of your screen. For example, the following tabs are choices appearing under the *Data* tab.

Map Navigation Buttons

Use the following buttons to navigate the map.

- **Elevator Buttons**: Below the map image and to the right of the *Mode* buttons are numbers starting with zero. These are elevator buttons. The number highlighted indicates the zoom level you are currently viewing on the map. The *Key Map* has a zoom level of *0*; the highest zoom level will show you the greatest detail on the map. Click *0* twice to center the *Key Map*.

- **Min/Max Buttons**: At the top of the bar between the map and the *Control Panel*, there are two small arrows at the top of the bar. Click the top arrow to maximize the *Control Panel* to full screen. To go back to the original screen, click the other arrow. Click the bottom arrow to maximize the map to full screen.

- **Mode Buttons**: Below the map image are three mode buttons, one of which is always selected. When you click on the map, the selected mode button controls the action.

- **Pan Buttons**: Around the outside of the map image are eight directional buttons pointed outward. Clicking these buttons will pan the view; the map slides in the opposite direction of the directional button.

Command Buttons

The following command buttons initiate action in the Map Viewer and the Map Viewing Configuration.

- **Add**: Click the *Add* button to add a snapshot to your report.
- **Apply**: Click the *Apply* button to apply changes so you can view them on the screen before they are saved.
- **Clear**: To remove the last item selected on the map, click the *Clear* button.
- **Clear All**: To remove all marked items on the map, click the *Clear All* button.
- **Delete**: Highlight the information that needs to be deleted and click the *Delete* button.
- **Exit**: Click the *Exit* button to close the program. You can also exit the program by clicking the in the upper right corner of the window.

- **Find**: To find the item highlighted, click the *Find* button.
- **Help**: Click the *Help* button to open the on-line version of this document, *User Guide: Map Translator Control*.
- **Hide**: For an item to be hidden on the map, highlight that item and click the *Hide* button.
- **Save**: To save your changes, click the *Save* button.
- **Show**: For an item to be visible on the map, highlight it and click the *Show* button.

Important

To save your changes, you must click on the *Save* button before clicking the *Exit* button. Clicking on the *Exit* button does not save your work.

Applying Changes

To apply changes and view them before they are saved, follow these steps:

- When all changes are made, move off the last change.
- Click the *Apply* button.
- Check the map to see if your changes are correct.
- To keep your changes, click on the *Save* button.
- To leave without saving changes, click on the *Exit* button.

Tip

Save your work after each time you have applied changes that are satisfactory.

Deleting Items

To delete an item, highlight it and click the *Delete* button. The following information is useful when configuring your maps:

- Information that appears as red is configured but not included in the data. If you delete this information, it will go away.
- Information that appears as blue is in the data but not in the configuration. This information cannot be deleted.
- Information that appears as black is both in the data and in the configuration. If you delete this information, it will reappear as blue the next time you open the Configure Map Viewing module.

Tip

To delete two or more consecutive items, hold down the shift key while clicking on each unit. To delete two or more items that are not consecutive, hold down the control key while clicking on each item.

Saving Changes

To save changes, click the *Save* button.

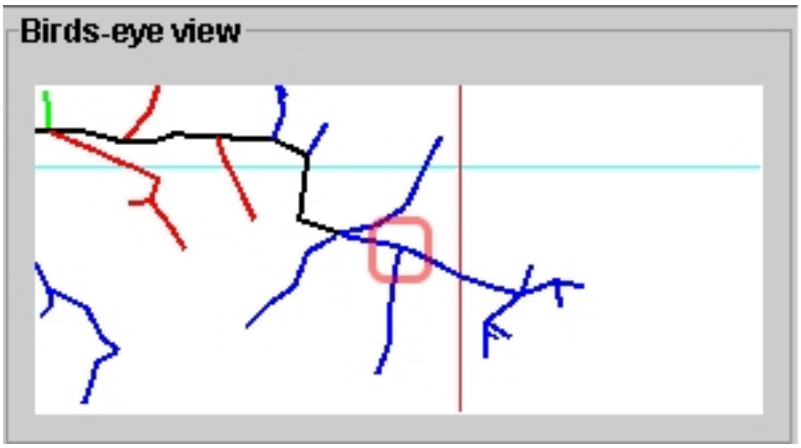
Important

Make sure you save your work often while working and save it again before leaving this screen.

Information at a Glance

The following features provide information at a glance.

- **Bird's-eye View:** When the Map Viewer is first opened, your company logo will be above the *Control Panel*. When you begin to zoom within the map, the logo changes to the bird's-eye view. Depending on your utility's specifications, the map in the bird's-eye view may be the same or a couple zooms below or above the map itself.



- **Coordinates**: The coordinates are visible in two places—above the map on the left side and above the table under the *View Data* tab. As you move your mouse around the map, the coordinates change. 476037, 1182986

- **Hover Indicator**: As you move your mouse across the map and hover over items, the most important information for the item over which you are hovering should be shown below the map directly above the *elevator* buttons. This information will be the information listed directly below the data type in the *View Data* table. 1/0QPX

- **Map Grid Numbers**: On the left side of the down pan button, the map grid numbers appear. As you move your mouse around the map, the numbers change to reflect the map grid numbers. 105-6-4 [This function will only be available if specified by your utility.]

- **Zoom Level**: Under the map on the right side, the *elevator* button that is highlighted shows you the

zoom level currently on the map. The higher the zoom level number, the more detail you will see.

Entering Information

You will encounter different types of entry fields when using the editing screens in the Map Translator Control.

- **Browse Box**: A pop-up box that appears when you press the *Browse* or *Add files* button, allowing you to select the path to the file(s) that need to be included in the configuration.
- **Picklist**: A box with a gray background and an arrow on the right side of the box. To change a picklist selection, click on the box and highlight your choice from the list provided.
- **Textbox**: A box with a white background for typing or editing information. To edit or insert information, click on the box and type in the new information.

Configuration Windows

Configuration windows provide screens for adding, editing, and deleting basic information about your maps. Different types of information are handled in different ways. Generally, the information will be handled by typing data into textboxes, selecting items from picklists, setting up directory paths through browse boxes, placing or removing check marks in check boxes, and selecting and moving information from one box to another by clicking an *Add*, *Add item*, *Include*, *Delete*, *Remove*, *Remove item*, or *Exclude* button.

Saving Configuration

To save your configuration changes, click the *Save Configuration* button at the bottom of the screen. We recommend saving often.

Exiting

To exit this application, click the *Exit* button on the bottom right on your screen. *Note*: Save before exiting. If you exit before saving your changes, you will lose all of your changes.

Help

When using the Map Translator, there are two ways to get *Help*.

- To get immediate help on a particular item on an editing screen, click on the textbox, check box, browse box, or picklist that needs edited. A description of that field will appear in the *Field Description* box on the screen.
- Click the *Help* button, and the index of this manual will appear.

Chapter 3. Getting Started

Goal

This chapter is a duplicate of the Quick Start guide included with the software. Its purpose is help you translate and view your maps with as little configuration as possible. Once you can actually see the maps and are familiar with the basic process, you should feel more confident proceeding with the rest of the configuration process.

Step 1: Install the Software

You will receive separate instructions for creating a Partner directory and unzipping the Map Translator to that directory. When correctly set up, you should have a Partner directory—for example, *C:\Partner*, which contains at a minimum the following subdirectories: ConfigManager, Maps, MapViewer, MapTranslator, Platform, and Shared. In the Partner directory you should find a document, *ReadMe.html*, which contains links to the manuals, in both HTML and PDF formats, for all installed Partner products. The HTML version of these manuals may also be accessed from the *Help* button within the Partner programs.

Note: Partner programs will run off a network drive, but using a local disk is faster.

The Map Translator Control program is *Partner\MapTranslator\resources\MapTranslatorControl.bat*. We recommend that you make a shortcut for this program on your desktop.

Step 2: Launch the Map Translator Control

Launch the Map Translator Control program. You will see a screen, resembling a web page, with several links. We refer to this as the Map Translator Control Panel, and each link executes a subprogram of the Map Translator.

Step 3: Configure Your Mapping System

From the Map Translator Control Panel, select *Specify Data Sources*.

Specify Data Sources Configuration Tool

There are six sections to this step depending on which type of maps you want to use with the Partner's Map Translator—ESRI ArcInfo SDE database, ESRI ArcInfo personal geodatabase, Autodesk AutoCAD, shapefiles, GE Smallworld, and U.S. Census TIGER maps. Please select and follow only the section(s) that pertains to your mapping system. The opening screen is shown below. Generally, data are entered into these fields either by typing information into textboxes, selecting items from picklists or browse boxes, or adding an item and changing its name. When you click on each data field, a description of that field appears below in the *Field Description* box.

When you are finished, click the *Save Configuration* button to save your work and then click the *Exit* button to leave this configuration tool. *Note:* If you want to save your changes, you must click the *Save Configuration* button before exiting.

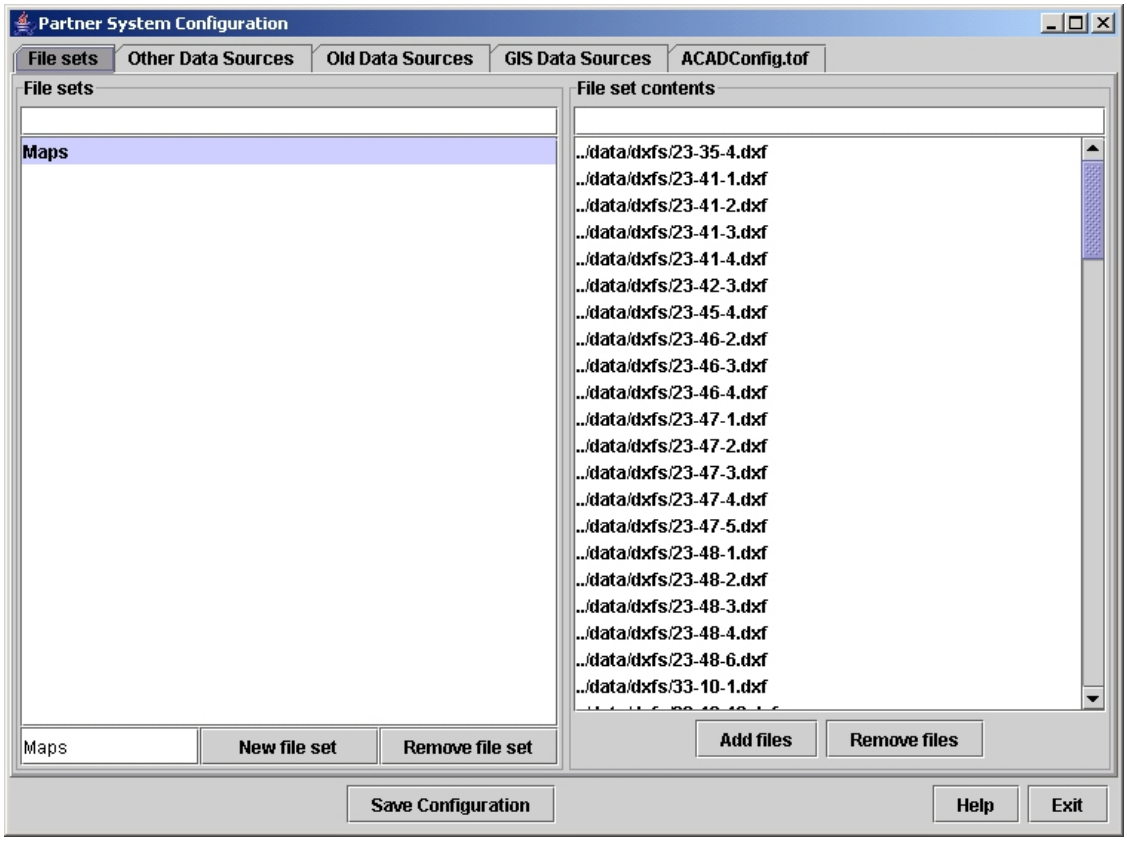


Figure 3.1. Specify Data Sources Opening Screen

Configure SDE Database Parameters

If your mapping system uses an SDE database, select the *GIS Data Sources* tab. Click on *ArcInfo frontend* arc in the left pane. In the right pane, enter the following:

- **Database type**—From the picklist provided, select the type of database to which to connect (generally SQL Server or Oracle).
- **Database server**—Type in the network address (name or IP address) of machine running the database to which you are connecting. This does not apply to ODBC connections. Use 'localhost' if the database program is running on the same machine as the Map Translator. If Oracle is being used, it may be necessary to specify a port number, usually 1521. Use a colon ':' to separate machine IP address from port (e.g., 10.10.10.10:1521). This also works with hostnames, such as MyOracleServer:1521.
- **Database name**—Type in the name of the database or ODBC source (usually "SDE").
- **Database user**—Type a valid user name for connecting to the database that has read access to all tables.
- **Database password**—Type a valid password for connecting to the database that has read access to all tables.

- **Personal geodatabase?—Make sure this is NOT checked.** If checked, the program treats the specified database as a personal geodatabase rather than an SDE database.
- Once all the properties are set, click the *Save Configuration* button and then the *Exit* button. You will return to the Map Translator Control Panel.

Configure Personal Geodatabase Parameters

Select the *GIS Data Sources* tab. Click on *ArcInfo frontend arc* in the left pane. In the right pane, enter the following:

- **Database type**—From the picklist provided, select Access.
- **Database server**—Leave blank.
- **Database file**—Click the *Browse* button and browse to a personal geodatabase. It will have a an .mdb extension.
- **Database user**—Leave blank.
- **Database password**—Leave blank.
- **Personal geodatabase?**—Make sure this box is checked.
- Once all the properties are set, click the *Save Configuration* button and then the *Exit* button. You will return to the Map Translator Control Panel.

Configure AutoCAD Parameters

If your mapping system uses AutoCAD, select the *ACADConfig.tof* tab. This configuration tool allows you to put in the path to AutoCAD. Set the following information:

- **Path to acad.exe**—Click the *browse* button and navigate to the folder containing *acad.exe,* which is usually somewhere like *C:\Program Files\Autodesk\ACADMAP5\acad.exe*.
- **AutoCAD 2000?**—Check this box If you have AutoCAD 2000 or later.
- **Windows Version**—Using the picklist, select which version of Windows you are using.
- Once this information is complete, click the *Save Configuration* button.

Select the *File sets* tab. This configuration tool allows you to select all the drawings for your Key Map and other map views. You have two basic file sets—Key and Maps.

- Highlight *Maps*; then press the *Add files* button.
- A browser box will pop up. Using the picklist provided beside *Files of Type*, select the type of AutoCAD file(s) you are using—either DWG or DXF—then select the appropriate file(s).
- Click the *Open* button.
- If you have a *Key* map, highlight the *Key* file set and repeat the instructions used for Maps.

- Click the *Save Configuration* button and then the *Exit* button.

Configure Shapefile Parameters

If your mapping system uses shapefiles, select the *File sets* tab. This configuration tool allows you to select all the drawings for your *Key Map* and other map views.

- Highlight *Maps*; then press the *Add files* button.
- A browser box will pop up. Using the picklist provided beside *Files of Type*, select *ESRI shapefiles* and then select the appropriate file(s).
- Click the *Open* button.
- Click the *Save Configuration* button and then the *Exit* button.

Configure GE Smallworld Parameters

If your mapping system uses GE Smallworld, you need to first extract the mapping information into XML files by following the steps in Appendix B.

After you have generated XML files, start the Map Translator and select the *File sets* tab. This configuration tool allows you to select all the XML files for your Key Map and other map views.

- Highlight *Maps*; then press the *Add files* button.
- A browser box will pop up. Using the picklist provided beside *Files of Type*, select *XML files from GE Smallworld* as the type of file(s) needed for your Key Map and then select the file name(s).
- Click the *Open* button.
- Click the *Save Configuration* button and then the *Exit* button.

Configure TIGER Files Parameters

TIGER maps are usually used in conjunction with existing maps. They can be used to add address range, street name, county, and water feature information to maps lacking in any of these areas. TIGER maps will only line up correctly with maps that are in a standard state-plane projection. To add TIGER files to your maps, follow these steps:

- Open www.census.gov/geo/www/tiger/tigerua/ua2ktgr.pdf and go to *Appendix A—State and Country Codes and Name* (page 187). Locate the state and county (or counties) in which you are interested and note the FIPS numbers for each county.
- Go to www.census.gov/geo/www/tiger/tigerua/ua_tgr2k.html and select the state. Using the FIPS numbers, download the zip files for each county. [If you need county boundaries, go to www.census.gov/geo/www/cob/co2000.html and download the shapefile version of the county boundaries for the state with which you are working.
- Using the Map Translator subprogram *Specify Data Sources*, select *GIS Data Sources*. Click on *tiger frontend*. Choose the projection you want for the Map Viewer.

- Select the *File sets* tab. This configuration tool allows you to select all the drawings for your Key Map and other map views.
- Highlight *Maps*; then press the *Add files* button.
- A browser box will pop up. Using the picklist provided beside *Files of type*, select the type of file(s) needed for your Key Map and then select the file name(s). Click the *Open* button.
- Click the *Save Configuration* button and then the *Exit* button.

Note: Website references here are only current as of the publication of this document.

Step 4: Adding Other Data Sources to Maps

If Partner Software previously configured another data source to include with your maps, you need to configure that data source for your specific machine.

From the Map Translator main screen, select *Specify Data Sources* (as you did for setting up maps). Select the *Other Data Sources* tab. If there are data sources already set up, they will appear in the left pane. For example, Partner will often merge external consumer data from a SQL database. In this case, there will be a consumer's entry in the left pane. In the right pane, there will be an *Index field* that is the field from the external data source that will match up to merge field (configured elsewhere) in the mapping system.

Set the *Database type*. If you set the database type to SQL Server, Oracle, AS400, ODBC or MySQL (not Access), enter the following:

- Database server—Type in the network address (name or IP address) of machine running the database to which you are connecting. This does not apply to ODBC connections. Use 'localhost' if the database program is running on the same machine as the Map Translator. If **Oracle** is being used, it is necessary to specify a port number, usually 1521. Use a colon ':' to separate machine IP address from port (e.g.,10.10.10.10:1521).
- Database name—Type in the name of the database or ODBC source.
- Database user—Type a valid user name for connecting to the database that has read access to all tables.
- Database password—Type a valid password for connecting to the database that has read access to all tables.

If you select a database type of Access, you will only need to click the *Browse* button next to *Database file*. Browse the Access database you want to include, and click *Open*.

Finally, you will need an SQL query. Partner Software usually creates this query for you. Writing SQL is outside the scope of this document, but there are many references on the web and in your vendor's documentation on writing SQL. For the Map Translator, it is best to use very generic SQL, avoiding joins and vendor-specific extensions.

To configure other types of data sources, consult the Partner manual. You may add CSV (comma-separated value) files, DBF files, Excel files, or SQL data sources (most databases) by selecting one these in the lower left corner of *Other Data Sources* and clicking the *Add new* button. You will get a configuration screen that differs based on type of data source selected in the right pane. For more information, please consult the Map Translator Control user manual.

Step 5: First Translation

(If Partner Software has previously configured maps for you, skip to Step 8.)

This is where the rubber meets the road. On the Map Translator Control Panel, select *Translate Maps*. The log tab will appear, and you will see a lot of messages. If they are mostly red, then you have a problem. Read the log; quite often it will tell you exactly what the problem is. Permissions are one of the most common problems (other than setting the properties incorrectly). If there aren't any errors, then all is well.

Step 6: Verify Results

Since Partner Software configured this map for you, you should be able to see you maps at this point. If not, you will need to look at the log files to find out what went wrong during translation. From the main Map Translator menu, click the *Zip Up Logs* link, which will create a file in the Partner directory called *Logs-[current date/time].zip*. Partner Software may ask you to send this file to us in the event you are having a problem.

From the Map Translator Control screen, run *View Maps*. The Map Viewer should come up with your maps centered on the screen.

Step 7: Proceed with Configuration

At this point you are translating your maps into the Partner Map Viewer format and viewing them as one seamless map with the correct center point. This is an essential milestone to configuring a map set.

There is plenty more to do, but you can proceed in any order. Feel free to play with the configuration tools if you feel comfortable with them or to consult the included manual (all the Map Translator tools have a *Help* button that brings up the manual).

We suggest you proceed with the following areas.

Configure Map Viewing

- line colors
- line types
- point icons
- text fonts and colors
- visibility
- zoom scales

General Map Settings

- background color
- foreground color
- printing background color

- centering point
- map extents (to eliminate extraneous data)

Configure Translator

- availability of features by LOD
- associated data field availability and labels
- find items
- labelling

Partner can help you with additional configuration, such as:

- adding additional external data sources;
- splitting line or point styles based on data contained in a field.

Your eventual goals should be:

- good looking, fast map set with appropriate zooms,
- uncluttered but informative display,
- appropriate associated data,
- find items for all desired searches.

Chapter 4. Translator Configuration

Introduction

At this point, you should have completed the *Quick Start* and have a working, partially configured Map Translator. This chapter describes how to complete the configuration. The Map Translator will not need to be configured every time you translate maps.

Read this entire chapter before you dive into configuration. You will move back and forth between sections as you refine the configuration.

Setting Background Color and Centering Point

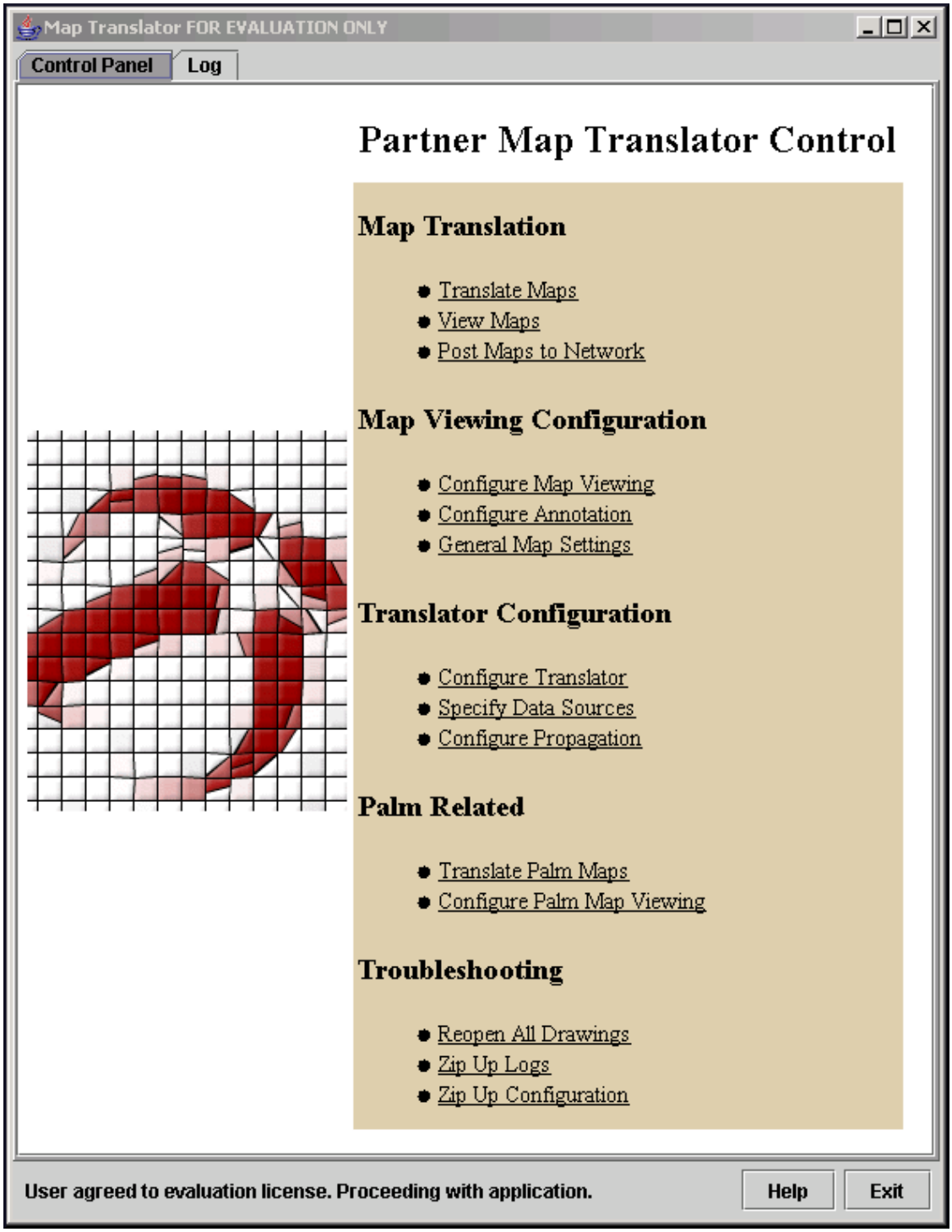


Figure 4.1. Map Translator Control Panel

The Map Viewer can have any background color (see Appendix A). Most installs use white, black, or lightGray. Set the background color before specifying the colors of lines, text, and points. The centering point controls the location of the *Key Map* when the Map Viewer starts.

To set the background color and/or the centering point, start the Map Translator Control and go to *General Map Settings*. Enter the background color. Enter a foreground color that will contrast with the background color (we use the foreground color for writing messages on the map). Save the configuration and exit.

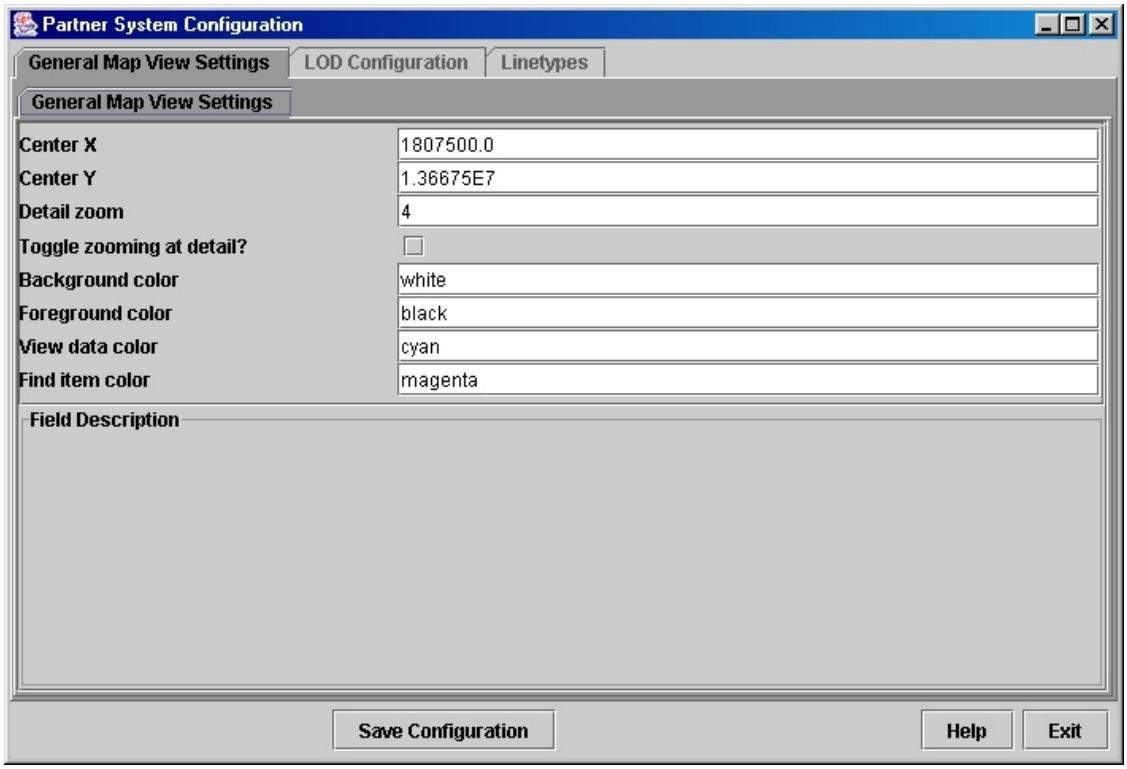


Figure 4.2. General Map Settings

To make sure the centering point is correct, open the Map Viewer. The map should be centered in the map screen. If it is not, go back to *Step 6: Set the Centering Point* in Chapter 3.

Controlling Scales and Visibility

Scales and visibility are controlled by the *LOD Contents* tab in *Configure Translator* and by the *Scaling* and *Visibility* tabs in *Configure Map Viewing*.

LOD Contents Tab

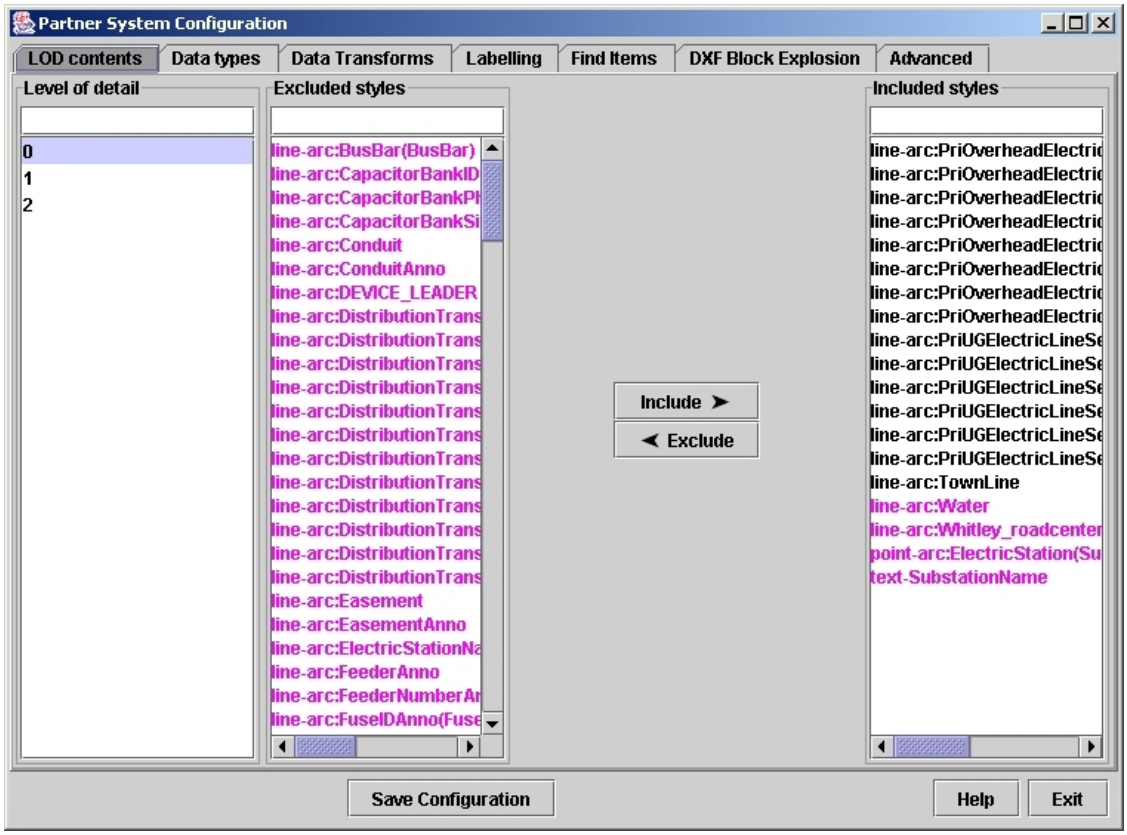


Figure 4.3. Configuring the Translator

The *LOD Contents* tab in *Configure Translator* controls what styles are written to each Level of Detail (LOD). For each LOD in *LOD Contents*, make sure all the styles that you want to be available in *Configure Map Viewing* are included in *Configure Translator*. After making and saving changes, translate your maps so those changes will show up in *Configure Map Viewing*.

Configuring Scales

The *Scaling* tab in *Configure Map Viewing* controls the number, scale, and birdseye of the zooms. For each zoom level, the *Scale* shows the scale in feet per pixels and indicates whether the bird's-eye view will be available at that zoom level. In the birdseye column, use *-1* to present a logo instead of the bird-seye view of the map.

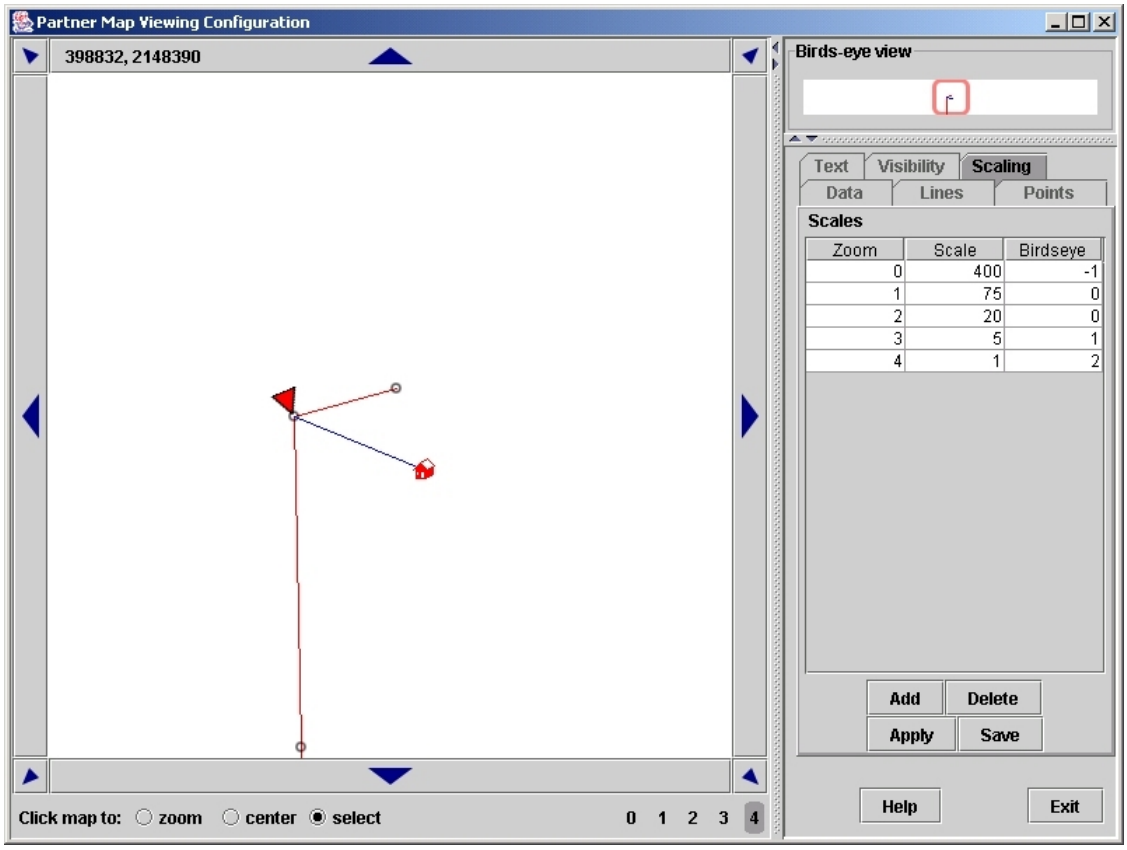


Figure 4.4. Configuring Scales

Configuring Visibility

The *Visibility* tab in *Configure Map Viewing* defines *Legend Items*, thus controlling which LOD is used and ultimately which styles are visible at each zoom level.

In the Map Viewer, each zoom level is defined by the *Legend Item* with the scale that is equal to or higher than the scale in the zoom level. For example, if you have *Legend Items* with scales of 1000, 100, and 20, then a zoom level with a scale of 30 will use the Legend Item with the scale of 100. Likewise, a zoom with a scale of 110 will use the *Legend Item* with a scale of 1000. Each *Legend Item* specifies the LOD to read and the styles from the LOD to display.

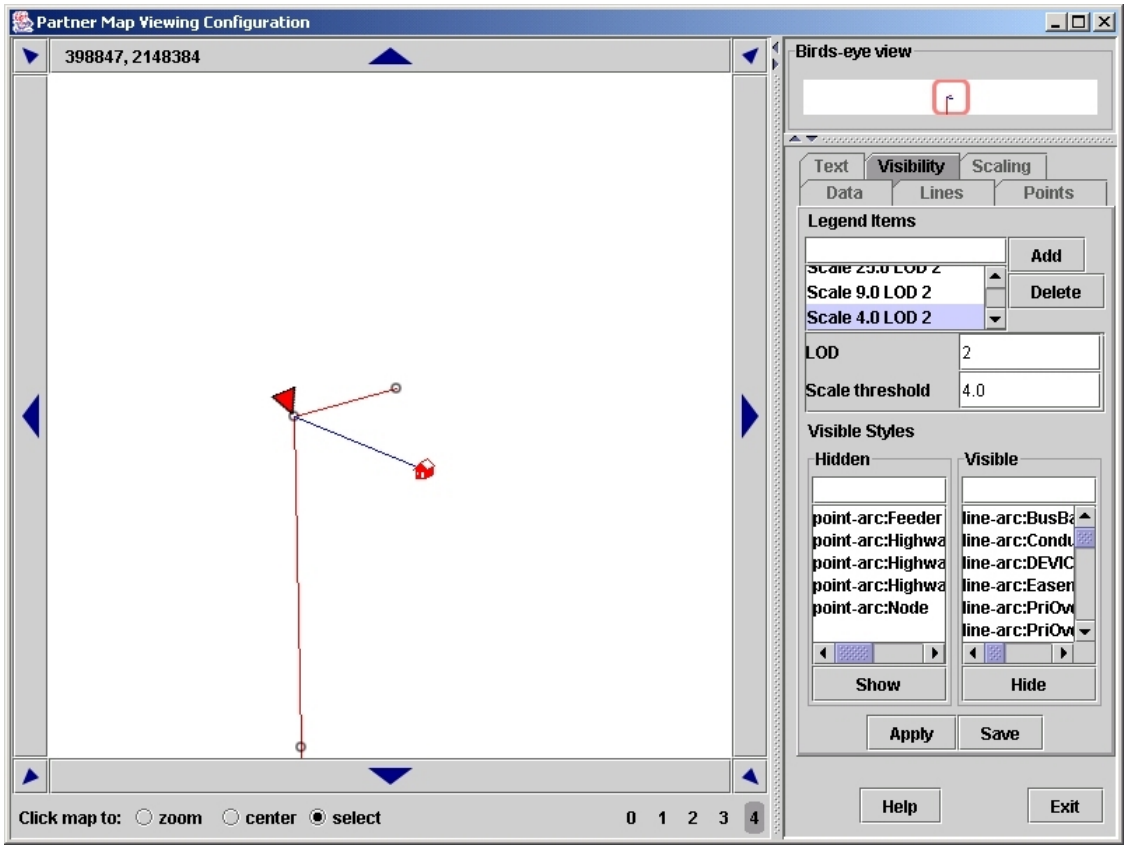


Figure 4.5. Configuring Visibility

Configuring the Key Map

Generally, you want the *Key Map* to fill the map screen. The map screen size is not constant because of varying screen resolutions and user decisions on how big to make the map screen. You can't make everybody happy but try to pick a scale that will fill the screen for the most common configuration.

Configuring Navigation Zoom Levels

We've found that most users want two *navigation* zooms below the *Key Map*. The purposes of these zooms include allowing the user to:

- drill down through these zooms to find their way to an area,
- use elevator buttons from a far-in zoom to these zooms to find his location,
- use these zooms for printing maps for directions,
- get an overview of the system for operations such as outage management.

Set the scaling of these zooms so that the user can comfortably zoom in from the *Key Map* and keep track of where they are. You'll need to play with the scaling until it feels right.

Generally, use level of detail (LOD) 1 as the source for the navigation zooms. Most people show roads, political boundaries, hydrology, subdivision lines, and primary lines. You may add control devices (reclosers, sectionalizers, etc) to the navigation zooms for the control center. Keep in mind that this can be a good example of less is more. If the Map Viewer responds slowly at these zooms, then you probably need to remove items from LOD 1.

Configuring the Finer Zoom Levels

The finer zooms use LOD 2 and present the detail of the maps. Typically, there are 1 or 2 zoom levels showing some of the data and 2 zoom levels showing all of the information. Some organizations have meter or other locations that have coordinates very close together. They create a final, very fine zoom to separate these items. The problem with this finest zoom is the user can lose his perspective and panning becomes nigh useless. We call this the "watching the ants crawl on the sidewalk" zoom. Also, consider how the printouts look.

You'll need to play with the scaling and visibility quite a bit to get something you like. Start in a very congested area and set a zoom that displays everything clearly. Then work out toward the navigation zooms, turning items off to keep the image clear. Go to different areas and make sure the maps are clear. Small changes in the scaling of zooms can make a big difference in how they look. You will probably find that your users will tell you everything they don't like about your configuration of these zooms.

Configuring Lines, Points, and Text

When your maps are translated the first time, all your lines will be red, all your text will be white, and all your points will be blue arrows named "Default". You will want making them look like your original map set.

Go to the Map Translator Control Panel and click *Configure Map Viewing*. Use this editing tool to control the appearance of the maps in the Map Viewers, including the color of your lines, the icons used to designate points, what you see on each zoom level, zoom level scales, and the color, size, and font of your text.

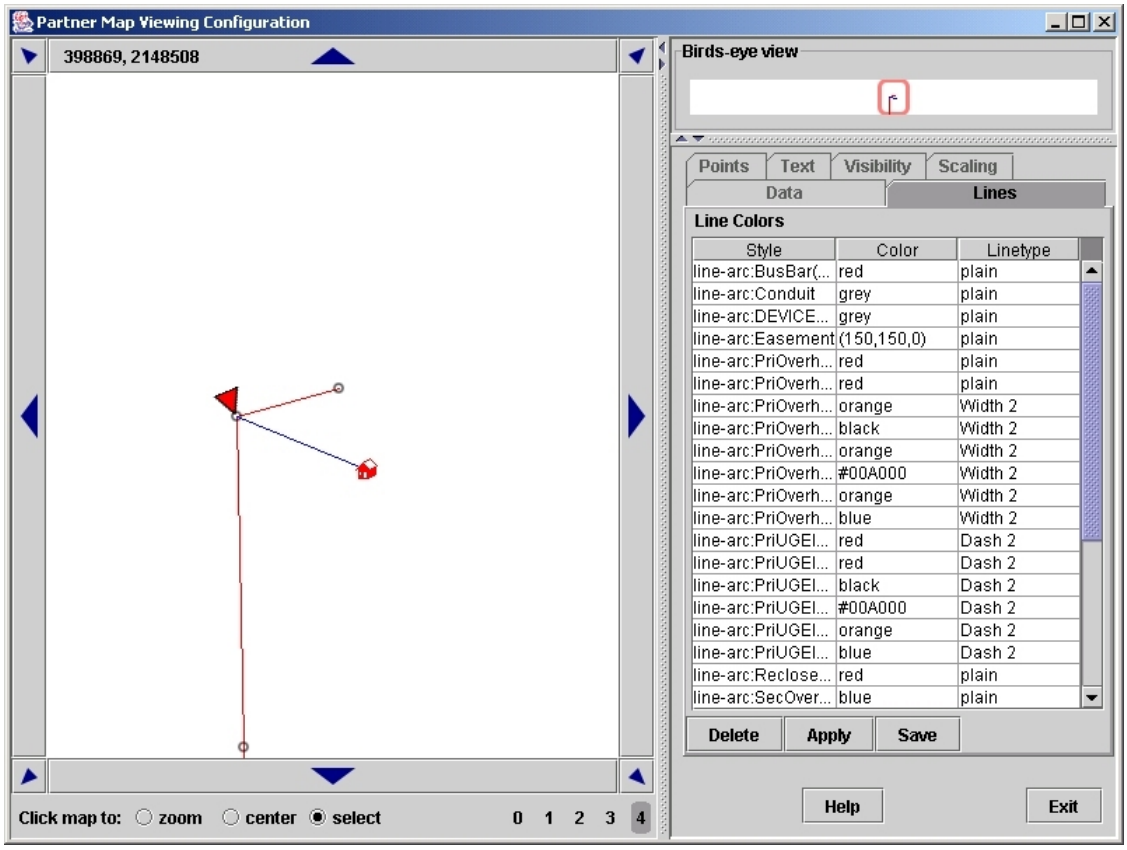


Figure 4.6. Configuring the Map Viewer

If you select the *Line* tab, you will see a similar editing screen where you can change the color of each line style. For more information on using colors, see Appendix A. From this window, you can also select line types, which are defined in *General Map Settings* window.

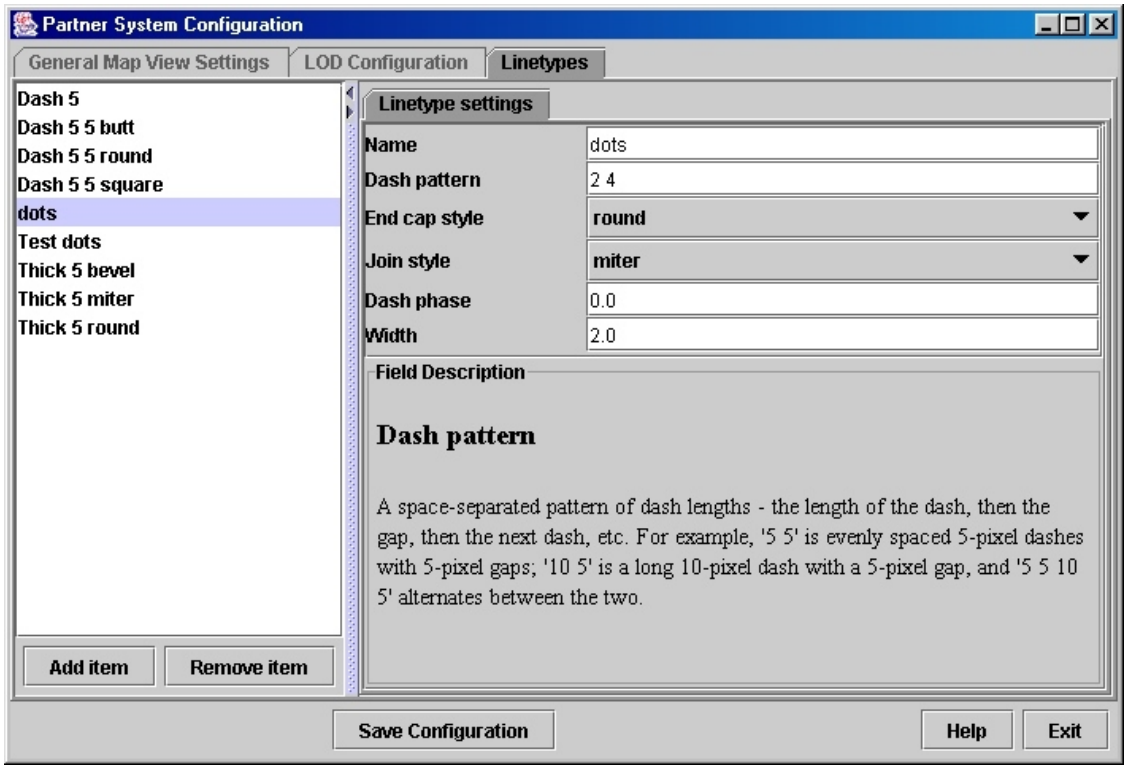


Figure 4.7. Configuring Linetypes

Look at the figure below. The *Text* tab is selected. Notice that a text style is highlighted. Double-click on the style and an example of that text style will appear on the map. Using this editing screen, you can change the type of font, size of font, and color of the text.

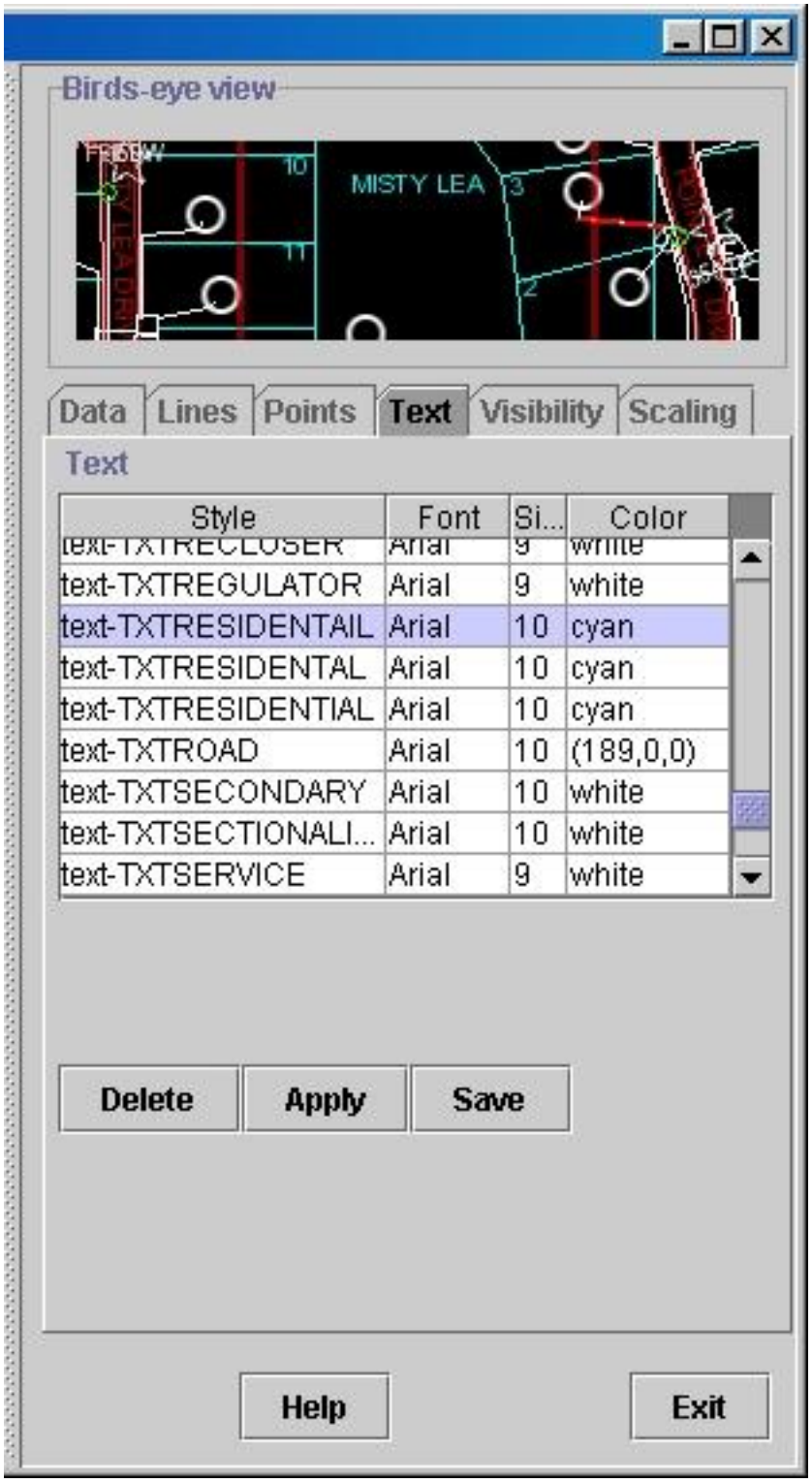


Figure 4.8. Configuring Text

To change the icons representing points, select the *Point* tab and click on any *Icon* box. A picklist will

appear, allowing you to select from a list of choices.

On each of the screens, follow these steps to immediately view your changes:

- Move off the last change you made.
- Click the *Apply* button.
- Double-click on the style you changed. An example of that style will appear in the middle of your map.
- Check this change at the different zoom levels.
- If you like this change, click the *Save* button. If you don't like the change, make appropriate changes and click the *Apply* button again.
- To leave this program, click the *Exit* button. *Note:* Be sure to save your work before pushing this button.

Configuring Data Types

The Map Viewer allows the user to view data about map entities using the *View Data* tab. The control of the viewable data, the order in the table, and the data labels are discussed below.

Getting Started

To configure data, start the Map Translator Control, go to *Configure Translator*, and then choose the *Data Types* tab.

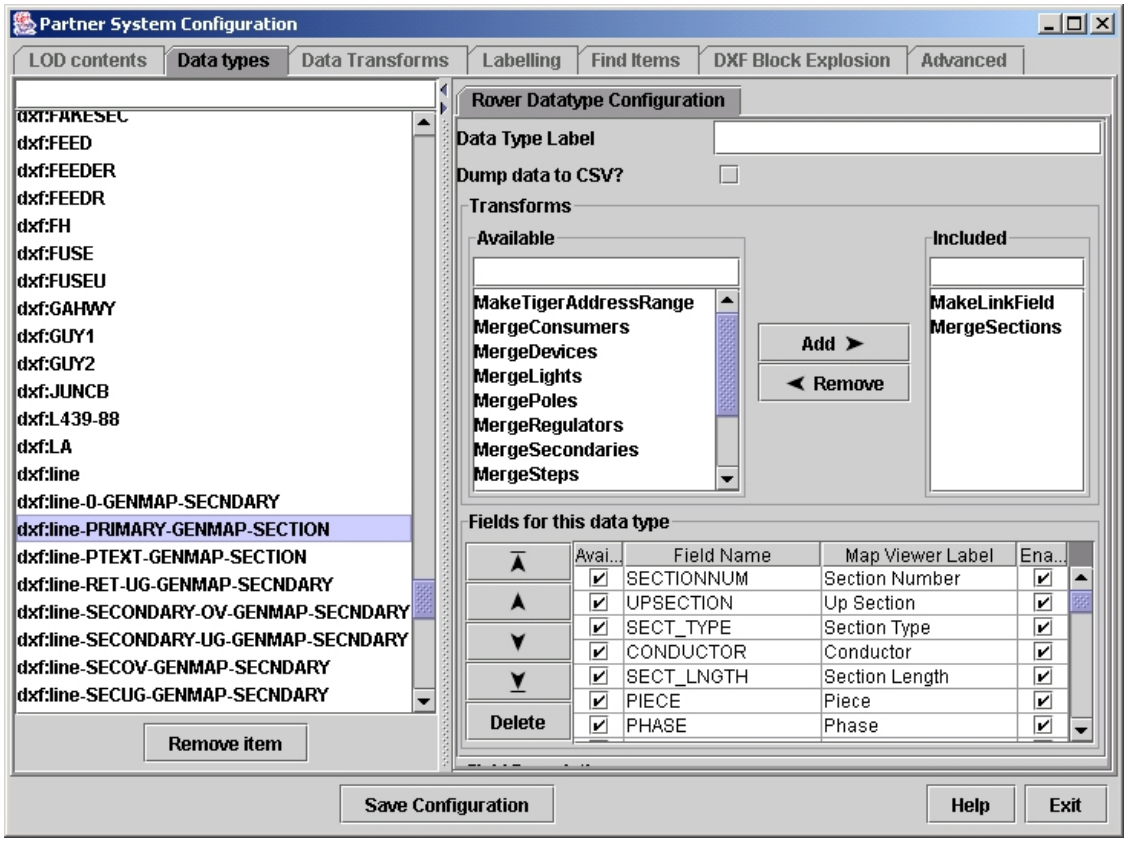


Figure 4.9. Configuring Data Types

Configuring Data Type Names

The column on the left lists the *Rover Data Type Names* identified in the map set. The syntax of the name varies between installations but in general the following style applies.

- At the start of the name, the data source can be present followed by a colon. You may see *dxf:* for data from AutoCAD drawings or DXF files, *shp:* for data from shape files, and *arc:* for data from ESRI databases. The data source is not present in many installations.
- For ESRI databases, the rest of the data name is the name of the table in the database and the subtype in parenthesis if the table has subtypes.
- For shape files, the rest of the data name is the name of the shapefile.
- For blocks from AutoCAD sources, the rest of the data name is the block name, a hyphen, and the Object Data table name (from the drawing file). Commonly, many blocks will not have attached object data in the drawings, so the name is just the block name.
- For lines from AutoCAD sources, the rest of the data name is *line-* and the object data table name (from the drawing file). All lines not attached to object data are listed as "line".
- For text from AutoCAD sources, the rest of the data name is *text*.

Click on the data type in the list with which you want to work. The selected data type will be highlighted.

Setting Data Type Labels

Type in a data type label for the data type. The data type label will be shown in the list of data types and will be listed above the data on the *View Data* tab in the Map Viewer. If you leave the field blank, the data type name is used as the data label. Reasons to add labels include:

- to shorten the data type name. Very long data type names are hard to read and make the window you are using hard to navigate.
- to rename data types to regular words. For example, you might have a data type CAPACITR that you label as Capacitor.
- to simplify the program. You may have many similar data types that you want to appear in the Map Viewer using the same data label.

Enabling and Sorting Data Properties

The available data properties are listed in *Property Naming* table on the right. The properties include data from the maps or tables, properties created during the Map Translation, and data from other sources merged to the data type (see adding data from other sources).

Enable the data properties that you want displayed in the Map Viewer by checking the box in the *Enabled* column.

Use the arrow keys to the left of the property table to move the data properties into the order you want them to appear when viewed. The property at the top of the list will be displayed on the down pan button as the end-user moves the mouse over items in the map.

Setting Map Viewer Labels

Change the labels in the column *Map Viewer Labels* to read the way you want your user to see the data.

Saving Your Work

Click *Save Configuration* to save your work.

Labelling

Labels are a powerful tool that generate text or point entities. You can add text labels to points or lines and control the position and rotation of each label. You also use labelling to create new point entities.

Getting Started

To configure labels, start the Map Translator Control, go to *Configure Translator*, and then choose *Labelling*.

Creating a New Label

Click the *Add item* button on the lower left. A new item named *no field set* will appear in the list. Click

on the *no field set*, and it will highlight.

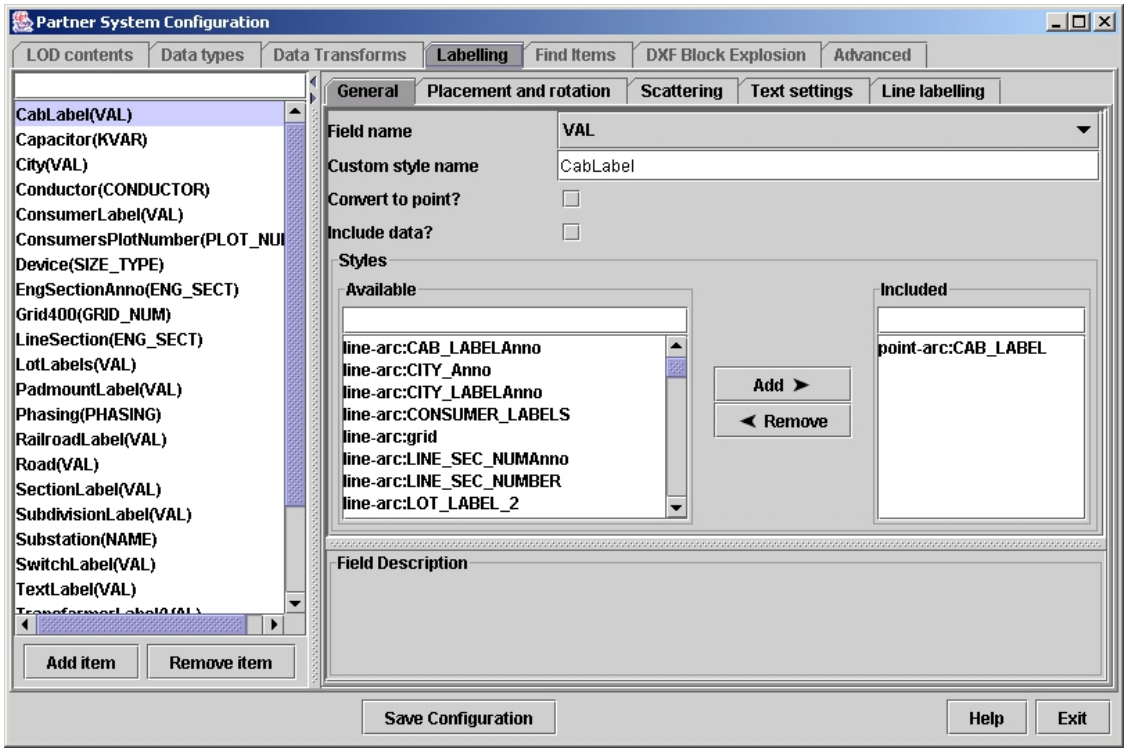


Figure 4.10. Adding a New Label

Naming Labels

You can either specify a label name or allow the program to apply a default name. The default is the data style name and the field name in parentheses. Use this default naming if you want to use a single label to generate many styles that can be configured individually during map viewing configuration. Specify a label name if you want a single style generated.

Selecting What to Label

Start by selecting the *Field name* to use as a label. For text to be added to the map, choose the field name for that text. For adding arbitrary text strings or points to the map, select any field name that exists in the entities you wish to label. After selecting the field name, the list of styles that contain that field name are listed as *Available*. Add the styles you want to label to the *Included* list.

Configuring Placement and Location of Labels

You control the offset and rotation of the label on the *Placement and rotation* tab. Help for each field is provided on the screen.

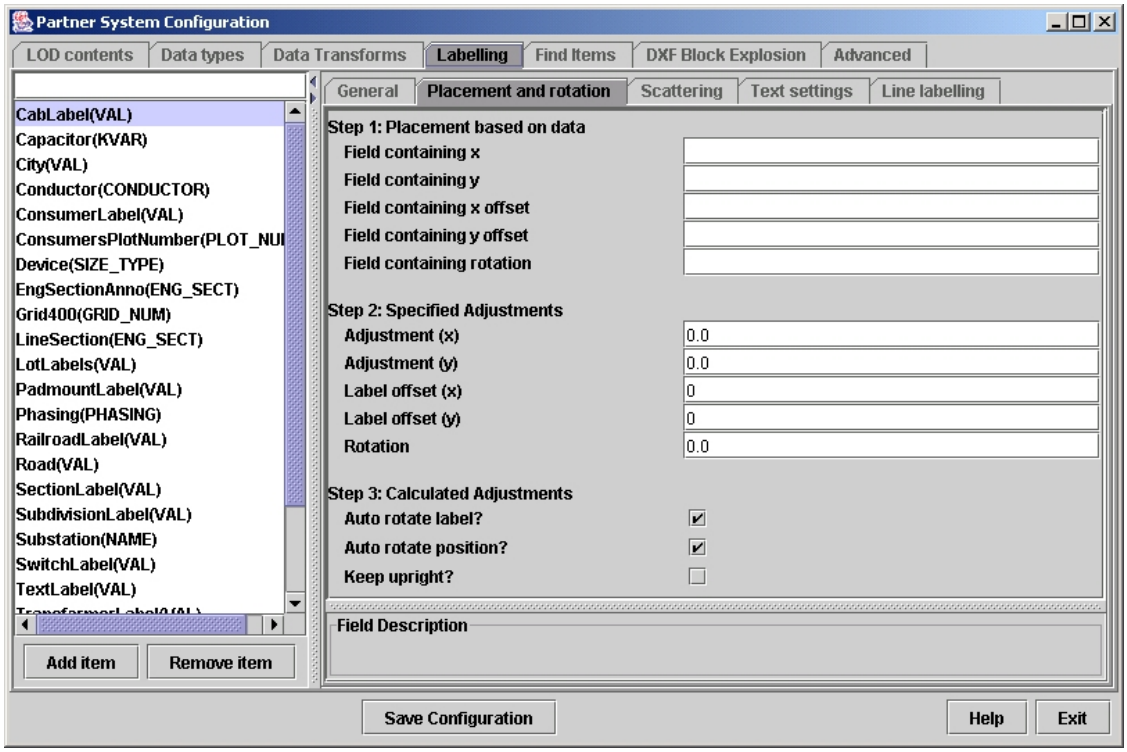


Figure 4.11. Configuring Placement and Location of Label

Configuring Label Text Settings

You control the text string and justification on the *Text settings* tab.

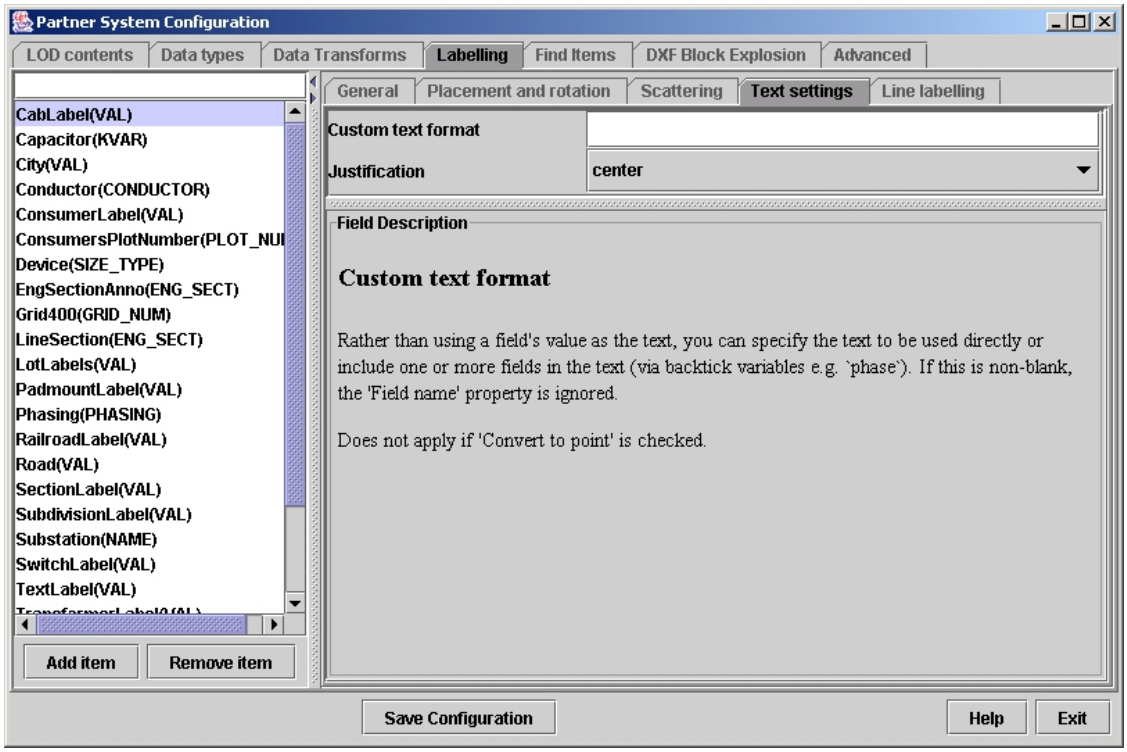


Figure 4.12. Configuring Text Labels

Line Labelling

You control the spacing and justification of line labels on the *Line labelling* tab.

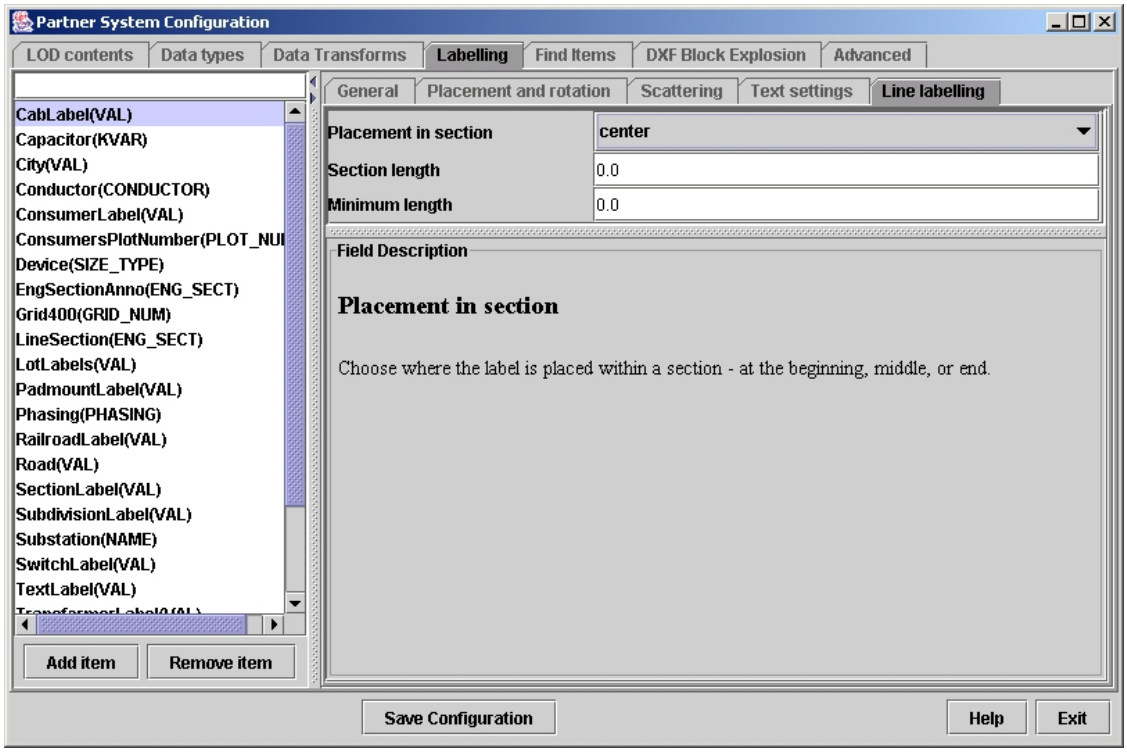


Figure 4.13. Configuring Line Labelling

Adding the Label to Translation

After the next translation, the new text or point entities will be listed in the *Excluded* column of the LOD contents. You need to move the label entities to the *Included* column and translate again to have the labels available under *Configure Map Viewing*.

Saving Your Work

Click *Save Configuration* to save your work.

Labelling Examples

Placing the KVA inside a Transformer Symbol

In this example, you want to put the KVA value as text inside the open triangle symbol used for overhead transformers. The icon rotates around an insertion point based on the actual placement of the transformer. At a zero rotation the transformer is due east of the insertion point. You want to place the KVAR value for the transformer inside the triangle.

- Create a label with the field KVAR chosen and the transformer style included.
- Offset the label 10 pixels in the *x* direction. It generally takes several tries to get the offset right.
- Specify a rotation of 90 degrees to rotate the text inside the triangle so the top of the letters is toward

the insertion point.

- Check the *Auto rotate position?* and *Auto rotate label?* so that the label moves and rotates to match the rotation of the transformer symbol.
- Set the justification to center under *Text settings*.
- Translate the maps, add the new text entity to the translation under *LOD Contents*, and then translate the maps again. Your label should be available under *Configure Map Viewing*.

Labelling Primary Line with Conductor and Phase

In this example, you want to label primary lines with phase and conductor. You need to set up two labels, one for each item.

- Create the two labels and select the fields and applicable styles.
- Offset one label 4 pixels in the *y* direction and the other label 12 pixels in the *y* direction. This will move the label insertion points above the line.
- Check the *Auto rotate position?* and *Auto rotate label?* so that the labels moves and rotates to match the rotation of the line.
- Set the justification to center under *Text settings*.
- Check *Keep upright?* so that the letters will not be upside down.
- Under *Line labelling*, choose a *Section Length* of 700 and a *Minimum Length* of 250. You will need to play with these values to get the labelling the way you want it.
- Translate the maps, add the new text entity to the translation under *LOD Contents*, and then translate the maps again. Your label should be available under *Configure Map Viewing*.

Labelling Primary Line with Flow Direction Arrows

In this example, you want to label primary lines with arrows indicating the current flow direction.

- Create the label and select any field present on primary lines and applicable styles.
- Check the *Auto rotate position?* and *Auto rotate label?* so that the labels moves and rotates to match the rotation of the line.
- You may need to rotate the symbol 180 degrees if the arrows point the wrong way.
- Check the *Convert to point?* box.
- Do not check the *Include data?* box. If this box were checked, every flow arrow would contain all the data for that section of line. This would make a much bigger map set.
- Under *Line labelling*, choose a *Section Length* of 400 and a *Minimum Length* of 150. You will need to play with these values to get the labelling the way you want it.
- Translate the maps, add the new point entity to the translation under *LOD Contents*, and then translate the maps again. Your label should be available under *Configure Map Viewing* as a point. Choose an appropriate arrow icon.

Configuring Find Items

The Map Viewer allows the user to find specific data items in the map using the *Find Item* tab. The control of the generation of these find items is discussed below.

Getting Started

To configure find items, start the Map Translator Control, go to *Configure Translator*, and then choose *Find Items*.

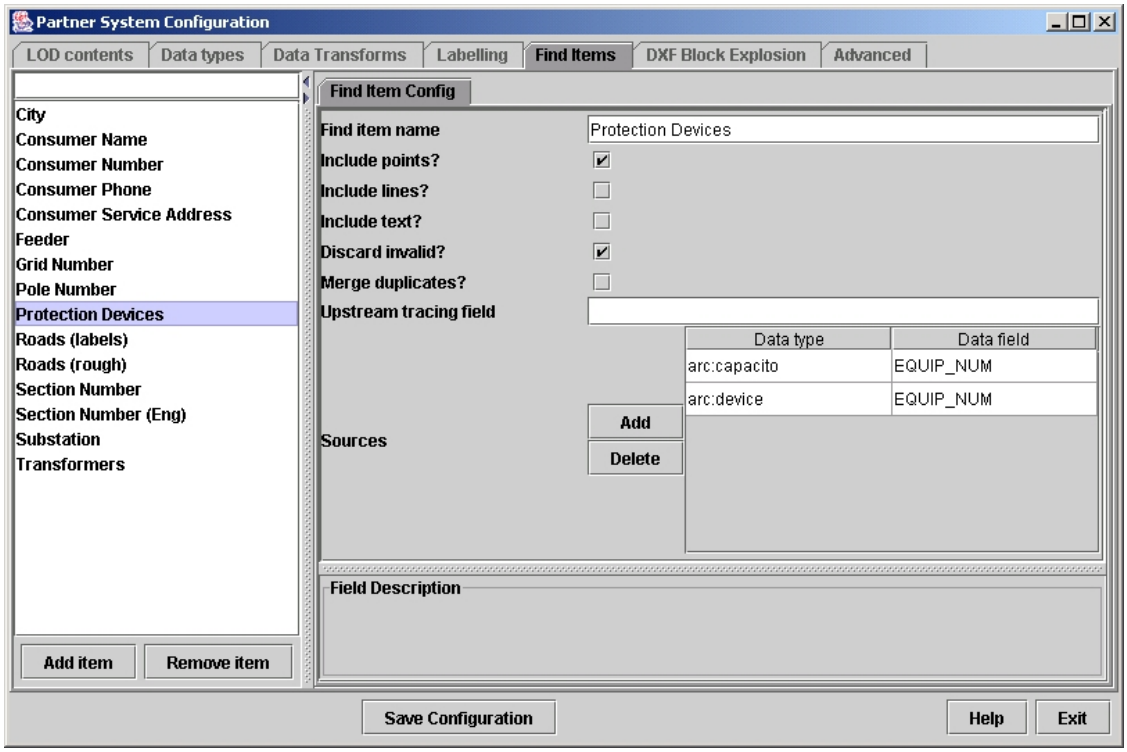


Figure 4.14. Configuring Find Items

Creating a New Find Item

Click the *Add Item* button on the lower left. A new item named *new find item* will appear in the list.

Click on the *new find item* and it will highlight.

Type the name you want to use for the find item in the box *Find item name*. This is the name that will appear in the list of find items in the Map Viewer.

Setting Find Item Conditions

Check the boxes next to points, lines, and text for the types of data you want included in the find item. You can use more that one type of data in the same find item.

Check the *Discard invalid* box if you want null or invalid entries removed. You may want to leave invalid entries in the data so that you can identify problems in your map set.

Check the *Merge duplicates* box if you want multiple entries merges into a single item in the find item list. Generally, you want to merge duplicates if you are using lines and not merge duplicates for points and text. If you merge duplicates, the Map Viewer will highlight all occurrences of the data and center of the first item encountered.

Selecting the Sources

Click *Add* to add a data source. A blank line is added to the sources table.

Click in the blank box in the *Data type* column. A list of available data types is displayed.

Select the data type you want added to the find item.

Click in the blank box in the *Data filed* column. A list of available data properties is displayed.

Select the data filed you want to use.

Add more data types if needed.

Saving Your Work

Click *Save Configuration* to save your work.

Removing Find Items

You can remove the configuration for a Find Item by using the Map Translator configuration tool. Removing the configuration will stop the translator from generating the Find-Item file during translation. However, the Find-Item file will still be present in the maps. To remove the find item, go to *Partner/Maps/resources/rover/find* and delete the *.dump* file with the name of the find item you want to remove. Partner does not automatically remove all find items during translation because find item files can be generated from sources other than the Map Translator.

Configuring Block Explosions

This section only applies to AutoCAD drawings and DXF files.

The Map Translator has the ability to convert the lines from AutoCAD blocks into lines objects in the Map Viewer. Exploded blocks look like the AutoCAD drawings. You can use labels to add text to these blocks. The point entity for the block is also available.

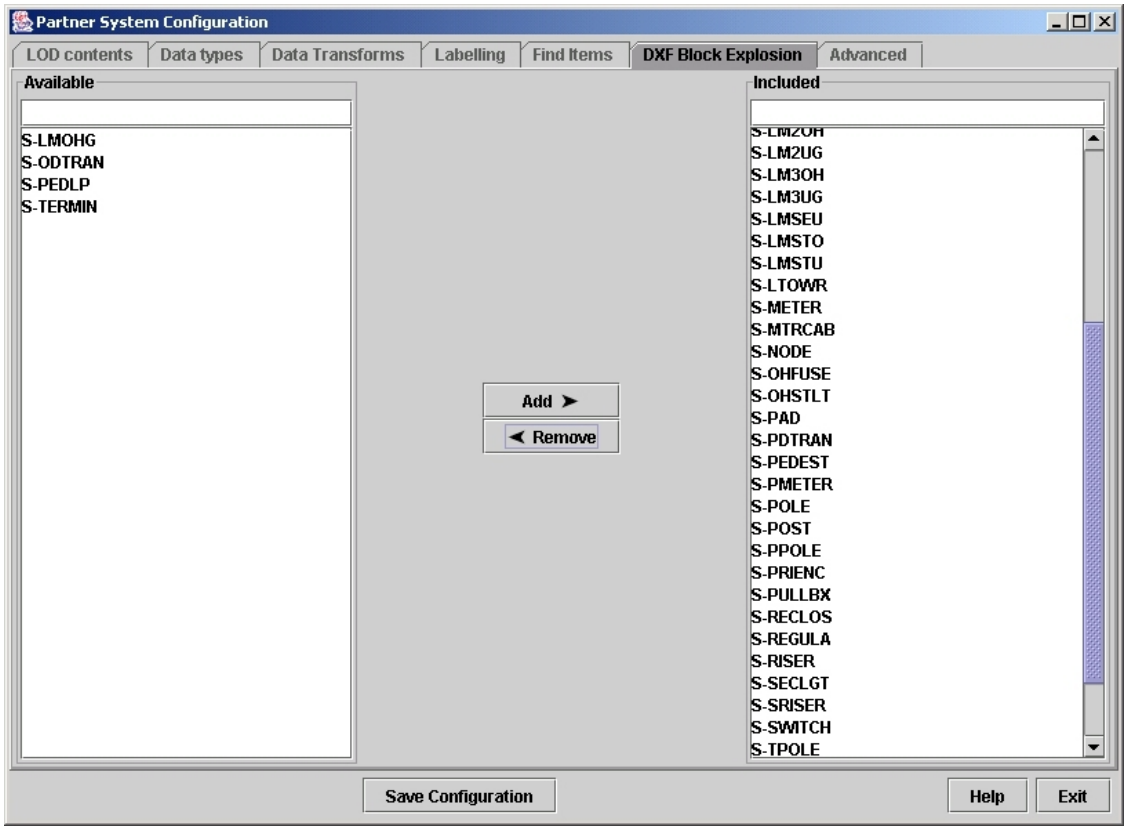


Figure 4.15. Configuring Block Explosions

The following are reasons to explode blocks.

- There are gaps in the lines if you use symbols.
- The block and the associated line objects in the drawing are closely related. You may have power lines ending right at the edge of a junction box symbol.
- The leader lines don't work well.

The following are reasons not to explode blocks.

- Exploding blocks increases the map set size.
- As you zoom, the object changes size, disappearing or taking over the screen.

You can have your cake and eat it too. Some clients explode blocks for the finer zooms but present icons for the higher zooms. They get the good relationships and the ability to see the object.

Merging External Data

The Map Translator can merge data from external, non-GIS sources for viewing in the Map Viewer.

Common examples of external data are customer information systems, inventory databases, and extensions to the GIS not directly included in map data.

The external data comes in as some sort of tabular form consisting of records with named fields. A combination of one or more of these fields must be chosen as an *index field* that will match up to a *merge field* in GIS data from the maps. After map translation is complete, data fields that successfully merged will be available to be made visible in the *Map Viewer*.

Configuration consists of: 1) identifying the index field(s) of the external data and the matching merge field of the GIS data; 2) creating a binary extraction (dump file) of the external data; and 3) matching the values of the index field(s) to those of the merge field through a Merge Transform. Where there is not a straightforward match, you can perform transformations on the data including padding with zeros, using coordinate or grid system naming to generate link fields, or manipulating the text to reorder or use only parts of the text for linking. See Python Transforms in this section for more information.

Types of external data sources include database tables, ODBC data sources, comma-separated text files, or fixed-block text files. Graphical configuration tools exist for SQL database tables, XML files, Excel files, and comma-delimited files (CSV). The SQL support includes generic ODBC data sources, as well as direct support for Access, SQL Server, Oracle, and IBM's AS400. Fixed-width files require special configuration beyond the scope of this document.

Creating a New Data Extraction (Dump File)

We assume that you have successfully translated your maps as described in the *Getting Started* chapter.

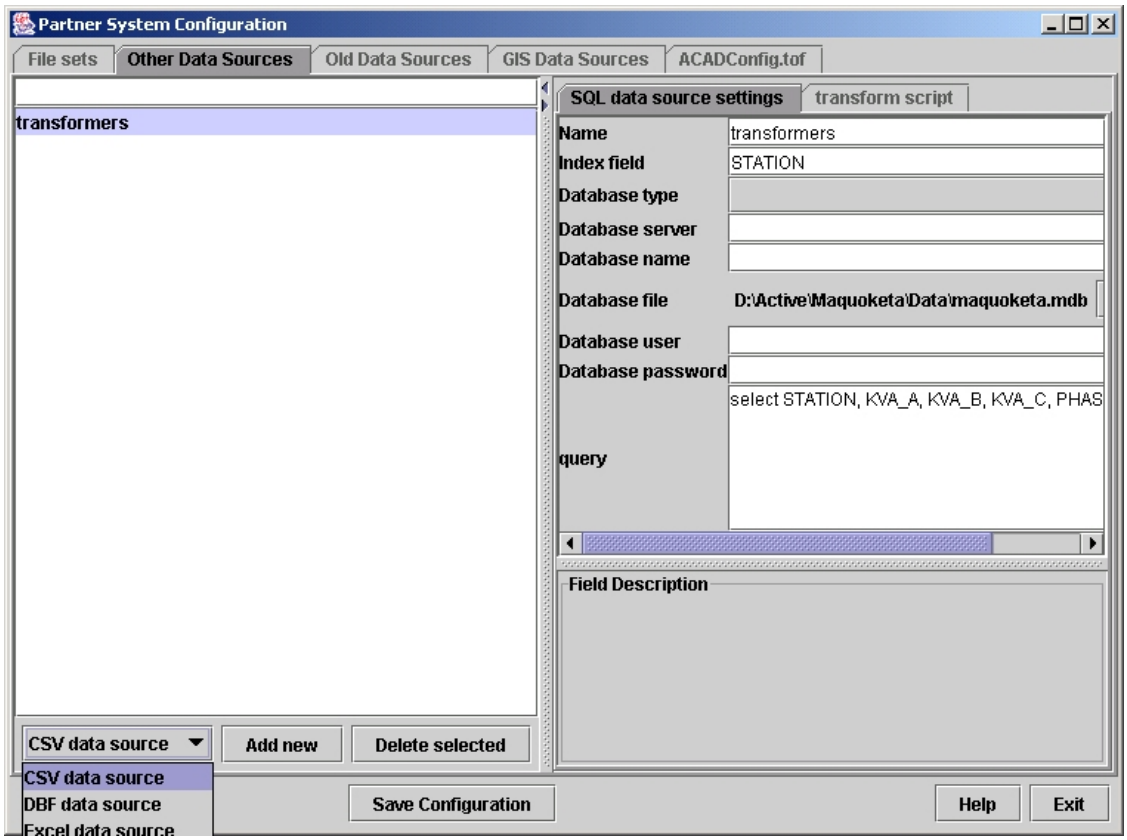


Figure 4.16. Configuring Other Data Sources

Start the Map Translator, go to *Specify Data Sources*, and then go to the *Other Data Sources* tab. From the picklist in the lower left screen, choose the format of your data from the following choices:

- **CSV data source**—comma delimited text files, such as those that can be exported from Microsoft Excel;
- **DBF data source**—Dbase files;
- **Excel data source**—a direct Microsoft Excel import;
- **SQL data source**—sources such as ODBC, Access, SQL Server, Oracle, and AS400 databases;
- **XML (Extensible Markup Language) data source**—some XML files can be read directly.

After selecting one of the above from the picklist, click the *Add new* button.

In the left pane, you should now have a new data dump item named according to its source type (e.g., CSVDump, SQLDump, DBFDump, etc.). Click this new data dump item to highlight it.

Change the default name for the data in the *Name* box. This name will be used to refer to these data later when merging the data, so choose a clear, memorable, and unique name. It should only contain alphanumeric characters with NO spaces (e.g., *ConsumerAccountingData*).

Then, at a minimum, the data source *Index field* must be identified. The *Index field* will merge with a corresponding *Merge Field* from the GIS data of the mapping system. It is critical that you specify a valid and properly spelled field as *Index field*. **Keep in mind**—the *Index field* for external data is set here; the GIS *Merge field* is set in *Configure Translator*, which is explained below.

For DBF files, Excel files, and XML files, click the *Browse* button to specify the path of the data source file. For SQL data sources, see the next section.

SQL Only: Specifying How to Read the Data Source

For an Access SQL database, choose SQL as the data type, then set the *Database type* field to Access. For ODBC SQL sources, specify database type as 'ODBC', and fill in *Database name* with the name of the ODBC source.

For all other SQL databases, you will need to specify the *Database type*, *Database server*—the hostname of the computer where the database resides, and *Database name*. If a username and password are required for database access, include those in the blanks specified. **Note:** Clicking on a field brings up a brief description of the required information. See Adding Other Data Sources to Maps for further assistance with merging Oracle and other external data.

Finally, SQL sources require a query statement, written in SQL understandable by the selected database. In most cases, this should be "select * from ExampleTable", which will extract all data from the table *ExampleTable*.

Sometimes, you will need to combine more than one field from the data source in order to make a link field to the GIS data. This can be done in the SQL statement directly. For example, to merge the variables SECTIONNUM and PIECE into a variable LinkField, include in the SQL " SECTIONNUM & '–' & PIECE as LinkField". This will put a dash between the two values in the LinkField variable.

Make sure you click *Save Configuration* before exiting.

For more information, consult a reference on SQL syntax such as your database documentation.

Adding a Merge Transform for the Data Source

Now that a data extraction has been successfully added to the translator in *Specify Data Sources*, you will need to complete the merge by telling the translator to what items in the maps you will merge data.

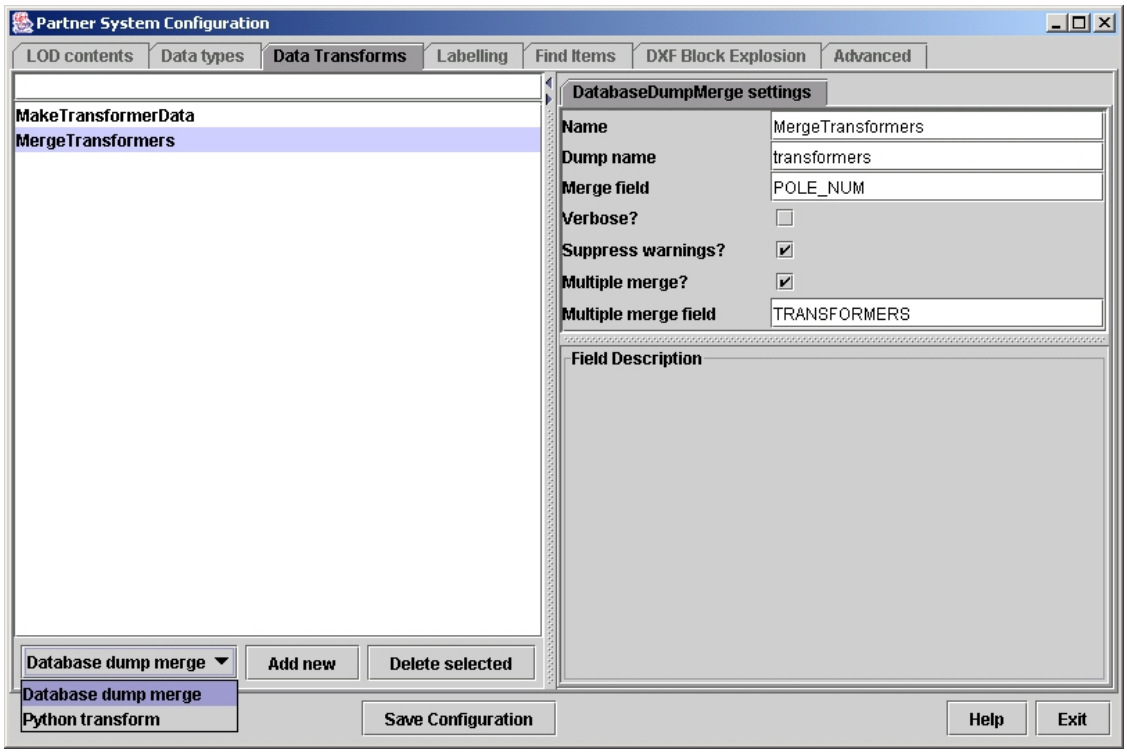


Figure 4.17. Configuring Merge Transforms

From the Map Translator Control, click on *Configure Translator*. Click the *Data Transforms* tab. In the lower left corner of the screen, select *Database dump merge* from the picklist. Click *Add new*.

A new merge item will appear in the Merge Transform list on the left. In the *Name* blank, give the Merge Transform a clear and memorable name, as you will use this later in the translator. It should only contain alphanumeric characters with NO spaces (e.g., *MergeConsumerData*). In the *Dump name* text-box, enter the unique name of the data extraction/dump file you created in *Specify Data Sources*. In the case of the above example, you would enter *ConsumerAccountingData*.

Enter the Merge Field. The *Merge Field* is the name of the field in the GIS data to use for merging. The data values in this field should have corresponding values in the index field of the dump file. It is critical that you specify a valid and properly spelled field here. This Merge Field name must match the original field name (not the Map Viewer Label) as shown in the *Data types* tab.

During map translation, data is first read from the external source(s) and then written to sorted binary files. As the GIS data is translated, the Map Translator will merge those files based on the *Index field* from the data source and the *Merge field* in the GIS data.

Click *Save Configuration* and then *Exit*.

Configuring the Map Translator to Merge the Data

Restart Map Translator Control, go to *Configure Translator*, then choose *Data Types*. New transforms, created above, only show up after a restart.

Select the data type you want to merge the external data with. For an AutoCAD system, the consumer data might be in 'dxf:consumer'; in ESRI it may be in 'arc:ServicePoint'. Select the data type in the left pane, look in the right pane for the transform you just created. Move the merge transform you just created from available to included. Save changes and exit. The maps must be translated at this point.

After translating the maps, run *Configure Map Translator* and click the *Data types* tab. Select the data type on the left, and look on the right under *Fields for this data type*. The newly merged fields will appear at the end of the field list and can be enabled and labelled just as you would any other data from the GIS system. You will have to translate one more time after enabling to see the data in the map viewer.

If the merged fields do not appear anywhere in the fields list, then the merge did not work properly. If you discover that the values of the index field and the merge field do not precisely match, then see the next section "Configuring Python Transforms" for a solution. Otherwise, consult the Troubleshooting section for assistance.

Configuring Python Transforms

If the values of the index field records in the map object are different from merge field records in the external data source (i.e., are arranged differently, have extra characters, etc.), then you will need to perform a Python transform on the index field in order for the merge to be successful.

For example, you wish to merge your accounting data with your mapping data in the Partner System. Your accounting data source contains a field named ACCOUNT, whose values correspond to those values of a field named METER in datatype 'arc:consumer'. Record '101234567890' of index field ACCOUNT corresponds to record '1234567890' of merge field METER. The values are similar only in that the last ten digits of the index field match those of the merge field. The first two digits of field ACCOUNT could be very useful on the accounting side but are problematic in the merge. Therefore, you need to remove the two accounting digits from the index field with a Python transform.

A Python transform in this context is a script written in the Python programming language which alters the values of the index field in the dump file to match up to the values of the merge field in the GIS data. Only the values in the newly created dump file are edited; the external data source itself is not affected. Refer to Chapter 8 for information on how the Partner System and Python interact.

To create a Python transform, open *Configure Translator* from the Map Translator Control menu. Click on the "Data Transforms" tab. Select "Python Transform" from the drop-down list on the lower left corner and click "Add new".

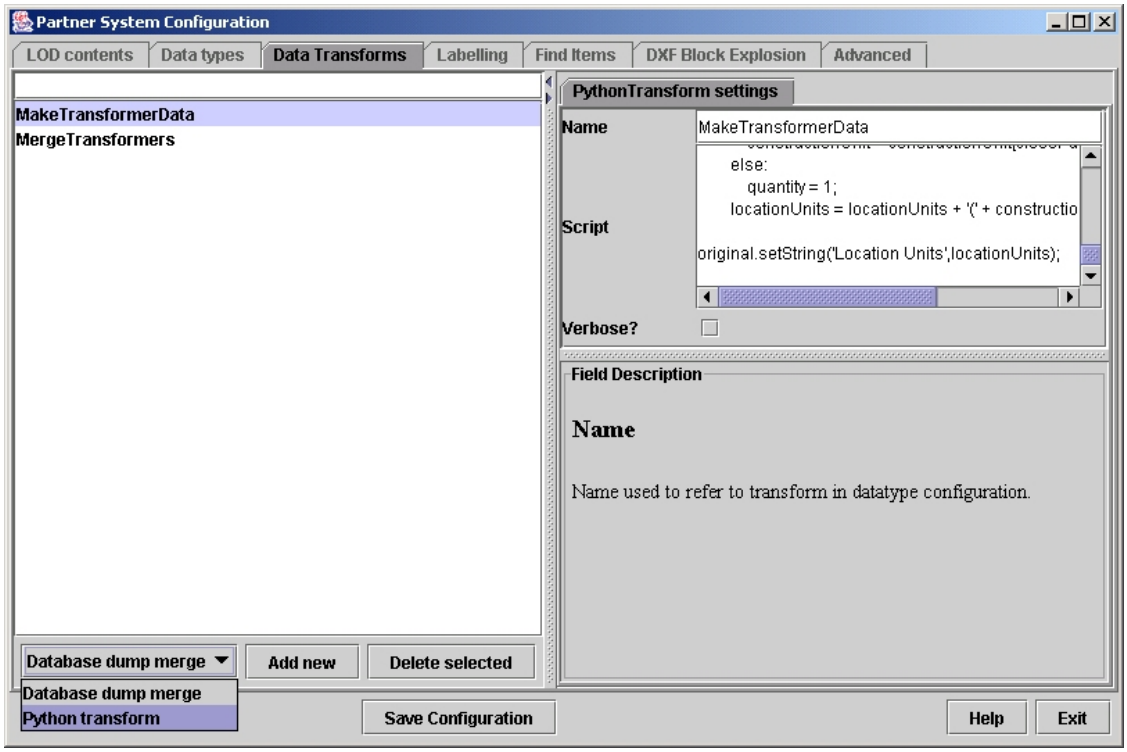


Figure 4.18. Configuring Python Transforms

A new item will appear in the merge transform list on the left; select to highlight and make active. In the *Name* textbox, give the Python transform a clear and descriptive name, one which illistrates its function (e.g., *MakeAccountIndexField*). The following is an example of a Python script that can be entered into the *Script* textbox:

```
AccountNumber=original.getString('Account');
IndexField=AccountNumber[-10:];
original.setString('IndexField',IndexField);
```

This script takes the last ten digits from field ACCOUNT to create the new field INDEXFIELD. For other Python script functions, see Chapter 8. When finished composing your script, click *Save Configuration* and then *Exit*. **Note:** It is necessary at this point to exit out of *Configure Translator* so the translator will recognize the newly-created Python transform upon re-entry.

Now we will apply the Python transform to the data type. In the case of the above example, we will apply transform *MakeAccountIndexField* to data type 'arc:consumer'.

Reopen *Configure Translator* and select *Data types*. In the data types list in the left panel, click *arc:consumer*. The configuration for this data type will appear on the right. There is a section called *Transforms* On the upper right side of the screen, you will see a section named *Transforms* with two lists underneath it—*Available* and *Included*. In the *Available* list, highlight the transform *MakeAccountIndexField* and click the *Add* button. Click the *Save Configuration* button, exit *Configure Translator* (and any other Partner application other than Partner Map Translator Control), and translate your maps.

The Python transform has been applied to the data type. Upon successful translation, the index field values will be changed to match the merge field values, and the external data will be merged with the GIS map data in the Partner system as configured in Merging External Data. Check to verify success of the

merge as shown in the previous section, new fields will show up at the end of the 'Fields for this data type." You will have to scroll to the bottom of the list to see the new entries.

Troubleshooting

If the merge is not successful, here are some troubleshooting tips:

- If you are having trouble getting Excel files to merge, convert them to comma-delimited (CSV) files.
- Double-check the spelling of both the index and merge fields, making sure that the punctuation and order of letters are correct.
- To confirm that database dumping is working, check the directory *MapTranslator/data/DatabaseDumps*. You should have three files with the data source name with extensions *.dump*, *.offsets*, and *.tof*. The files should not be empty. The *.tof* file lists the variables included in the extraction. The *.dump* file contains the actual data, though it is only somewhat human readable.

Configuring Propagation

Go to the Map Translator Control Panel and click *Configure Propagation*. Use this configuration tool to define each propagation package. The editing screen is shown below. Generally, data are entered into these fields either by typing information into textboxes, selecting items from picklists, or adding an item and changing its name. When you click on each data field, a description of that field appears below in the *Field Description* box.

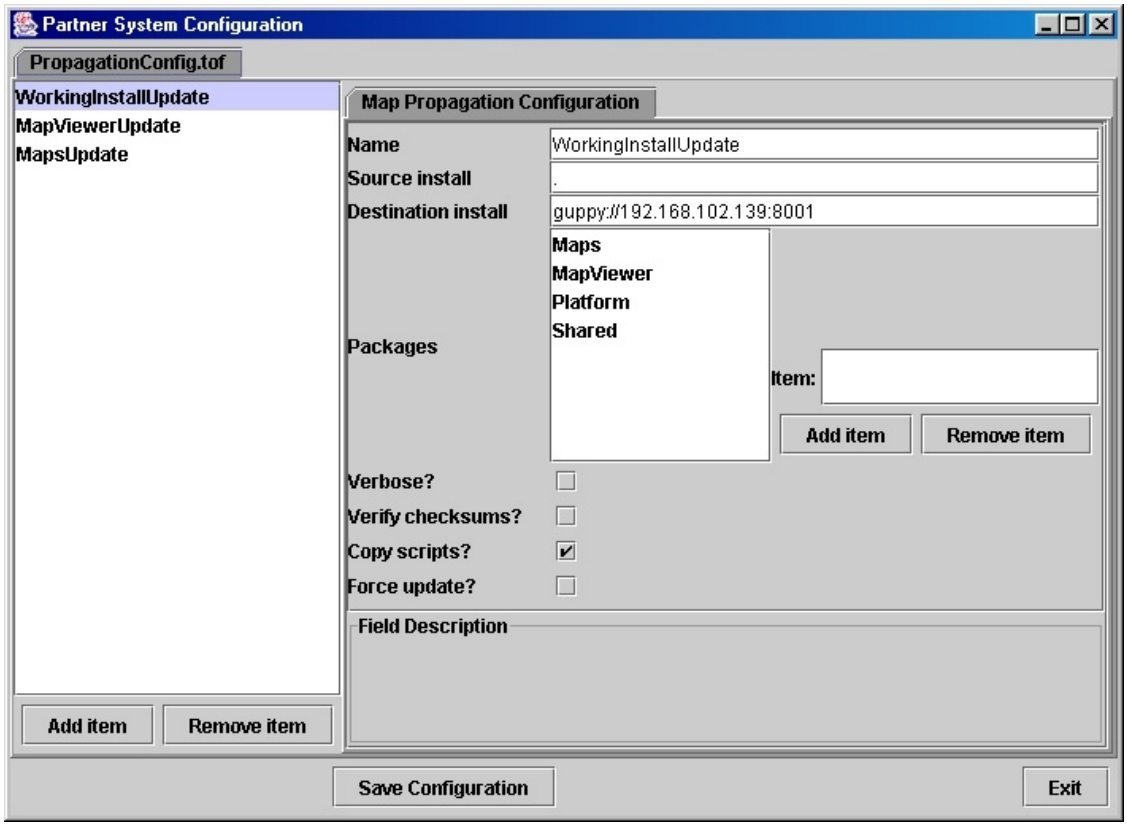


Figure 4.19. Configuring Propagation

When you are finished, click the *Save Configuration* button to save your work and then exit. *Note:* If you want to save your changes, you must click the *Save Configuration* button before exiting.

Configuring Partner's Palm Map Translator

Requirements

You will need a Palm OS®-compatible handheld with a secure digital (SD) card slot. The most recent Palm, Sony, and Garmin® units that have an SD card slot should run the Palm Viewer. Partner has tested most extensively on the Palm Tungsten™ T series and the Garmin® iQue™ 3600, which is an excellent integrated Palm/GPS unit.

Update your translator to the latest stable roll for best results.

Your Map Translator Control Panel may need to be separately updated. You should have links *Configure Palm Map Viewing* and *Translate Palm Maps* on the main screen. If you don't, call Partner Software.

Configuring the Palm Translator

- The Palm's screen is much smaller, and configuration should be done with this in mind. The key map will need to be scaled down—try to set your scaling (on the scaling tab) so that Zoom 0 fits into a 2-inch square on your screen.
- Currently, only a limited subset of icons is available on the Palm. Therefore, you must use the set of icons available on the Palm instead of any custom icons you may be using. Icons that are not available on the Palm will display as green question marks.

1. Start the Map Translator.
2. Click *Configure Translator*; go to the Advanced tab. Set the *Result Projection* to a projection. This should be the State Plane/UTM projection in which your maps reside. If your maps are not in a listed map projection, select a projection anyway because the Palm translator will not translate until this field is set. Click *Save Configuration* and exit.
3. Click the *Configure Palm Map Viewing* link. This launches an application that looks exactly like *Configure Map Viewing*. All the controls for Palm configuration are exactly like those for configuring the regular Map Viewer legend. However, keep in mind the following:
4. Save your changes and exit *Configure Palm Map Viewing*.
5. Click *Translate Palm Maps*. This will be similar to a normal map translation, except that it just creates the Palm maps. The Palm translator will create a new directory called *PalmMaps* in the top level of the Partner system.
6. Copy the rover directory contained in *PalmMaps\resources* to an SD card.
7. The Palm Map Viewer is located in *MapViewer\resources\palm*. Use the Palm HotSync® software to install this file onto the Palm. You may also run the Palm Map Viewer directly from the SD card by installing the *.prc* file to the *PALM/Launcher* directory.

Chapter 5. Maintenance & Production

Now that your Map Viewer is set up, managing the changes will be your challenge. We recommend the following process.

Daily Routine

Every time you update your map set or change your map viewing configuration, you need to translate the maps, view them, and post them to the network. Go to the first option on the Partner Map Translator Control Panel and follow these steps.

Step 1: Translate Maps

On your Map Translator Control Panel, click *Translate Maps*. The Map Translator software now reads the GIS and other information and creates the Partner map files.

Step 2: View Maps

On your Map Translator Control Panel, click *View Maps* to open a Map Viewer. Always check the maps to make sure your changes were made before posting them to the network.

Step 3: Post Maps to the Network

On your Map Translator Control Panel, click *Post Maps to Network* to calculate all checksums and upload the maps to the network so that all Map Viewers can be updated. A log screen will appear to provide necessary information.

Reconfiguration

Now that you have looked closer at your maps, you would like to make some changes. Read Chapter 4. Each time you make changes to your configuration or maps, follow the *Daily Routine* described above.

Installing and Updating Map Viewers

Installing the Map Viewer from a CD

Your installation package for the Partner Map Viewer should include a disk and user guide. Assumptions: The CD includes all the maps and necessary program files for the Map Viewer.

1. Copy the Partner subdirectory from the Map Viewer disk onto your local or hard (e.g., C or D drive; whichever drive has the most space) of the individual computers where the Map Viewer will be installed.

2. Double-click on the *RemoveReadOnly.bat* file located in the Partner subdirectory.

3. Create icons for the *Update.bat* and the *MapViewer.bat* files. To create an icon, highlight the file name, right click your mouse, select "Create Shortcut", and drag this new file to your desktop.

4. For Windows 98 users only: Right click on the *Update* icon and select "Properties". Under the "Program" tab, click in the box next to "Close on exit" to insert a checkmark.

Installing the Map Viewer from the Network

Assumptions: Partner Software has already installed all the maps and necessary program files on the network.

1. Create a new folder named, *Partner*, on your local or hard drive (e.g., C or D drive; whichever drive has the most space).
2. Copy the *Platform* subdirectory from the Partner Working Install on the network machine to the newly created *Partner* folder.
3. Run the Update.bat file located in the new *Partner\Platform\resources* folder to open the installation application. Click *Program and Maps* to download the Map Viewer program and maps from the Map Translator Control. While running, it launches a log screen with an *Exit* button.
4. Create icons for the *Update.bat* and the *MapViewer.bat* files. To create an icon, highlight the file name, right click your mouse, select *Create Shortcut*, and drag this new file to your desktop.
5. For Windows 98 users only: Right click on the *Update* icon and select "Properties". Under the "Program" tab, click in the box next to "Close on exit" to insert a checkmark.

Updating the Map Viewer

To update your Map Viewer, double click on your Update icon.

Running the Map Viewer Directly from the Network

You can run the Map Viewer directly from the network by installing a shortcut on your computer. However, this needs to be a mapped drive, it cannot just be a network neighborhood.

Automatic Translation: Configuring Windows Task Scheduler to Translate Maps

The following relatively simple steps show you how to run the Map Translator from the Windows 2000 Task Scheduler. The process for doing this from Windows NT or 98 is similar but will not be covered in this document.

These steps should be run on the machine that Partner Map Translator is installed on.

Note: Translation may be customized for many customers; if any customization needed to be done for your translator, you will require a custom batch file that matches the steps specified in the HTML code behind the Translate Maps selection of the Map Translator Control. In particular, you translator may require more memory to be avaiable than stock. This is done on the -mx64M command below. Try making this number as high as practical, ie, -mx512M would allocate 512 MB of memory to the translator. The generic translation batch file discussed below will work for most customers.

- 1) Verify that the user that the scheduler will run as will have access at the scheduled time to all the data and mapped drives required for translation - otherwise translation will fail.
- 2) From the Windows 2000 Control Panel, there should be Scheduled Tasks folder. Inside this folder, run the program "Add Scheduled Task." This will start the scheduled task wizard.
- 3) Click next until you get to selecting a program for Windows to run. Use the browse button to loc-

ate the Partner folder. Inside the Partner folder, find the file MapTranslator\resources\tools\RoverTranslator.bat.

- 4) Follow the rest of the prompts, deciding whether you want to run this daily, weekly, or even every other day. You should make sure to have this run at a time when the translator will have to all the data it needs to complete successfully.

- 5) The program runs as a user. Make sure this user has all required mapped drives (P:, etc) with the authorization to access all the needed data (maps, accounting, etc).

- 6) That's it. The translator will run when the Windows scheduler starts it. Someone will need to check out the maps before they are made avaiable to all users.

Note 1: Posting Maps to Network SHOULD NEVER BE AUTOMATED! Someone should always look at the maps (and a find item) to make sure everything is there before Posting the Maps to the Network.

Note 2: You may substitute MapTranslator\resources\tools\RoverTranslator-LowPriority.bat in the steps above. This runs the translator at low priority and will make the computer more usable for other tasks during translation.

Below is an example of the file Partner\MapTranslator\resources\tools\RoverTranslator.bat.

```
rem Partner Software Generic Program Launcher
rem ###############################################################
rem ###############################################################
rem
rem This program is designed to run from the MapTranslator\resources\lib\
rem directory. It will not work otherwise.

cd ..\..\..\

rem
rem ###############################################################
rem The actual program launcher
Platform\resources\jre\bin\java -classpath Platform\resources\lib\platform.jar
  com.PartnerSoft.platform.LaunchJava RoverDatabaseDumper

Platform\resources\jre\bin\java -classpath Platform\resources\lib\platform.jar
  com.PartnerSoft.platform.LaunchJava GISDataExtractor

Platform\resources\jre\bin\java -mx64M -classpath Platform\resources\lib\platform.
  com.PartnerSoft.platform.LaunchJava RoverTranslator

Platform\resources\jre\bin\java -classpath Platform\resources\lib\platform.jar
  com.PartnerSoft.platform.LaunchJava MicroRoverTranslator

rem ###############################################################
rem end of program
rem ###############################################################
```

Automated Email Notification

Components

The Map Translator can be configured to send emails. The most common implementation is to send emails to users when the maps are posted to the working install.

The configuration to send emails consists of:

- add an entry in the Map Control window configuration specifying to send email;
- create an entry in javaApps.txt calling a Python script that builds and sends the email;
- create the Python script that builds the email.

Configuration Details

The Map Control window is configured in Partner/MapTranslator/config/preferences/TranslationControlGUI.html. You need to add a call to a javaApps entry to this configuration. In the example below, the call to "SendNotice" has been added to the Post Maps to Network step.

```
<li><a href="#app-CalculateAllChecksums,MapPropagation,SendNotice"
><font color="#000000">Post Maps to Network</font></a>
```

Add an entry to Partner/platform/config/preferences/javaApps.txt for "SendNotice". The example below calls a Python script Partner/MapTranslator/config/preferences/SendNotice.py.

```
SendNotice com.PartnerSoft.tools.Python MapTranslator/config/preferences/SendNotic
```

Configure the specified Python script to send an email. The example below is such a configuration.

```
from com.PartnerSoft.util import SystemLog;
log = SystemLog.singleton();

import smtplib;
server = smtplib.SMTP('mail.partnersoft.com')
toList=["wfay@partnersoft.com"]
fromPerson=["wfay@partnersoft.com"]
message = '''Subject: E-mail from within Partner
This is a message that is being sent from the Map Translator on Bill's desk.
This is really easy
to set up.  Give me a call and we can make it happen. '''
server.sendmail(fromPerson,toList,message)
server.quit()
```

You need to specify the server, the fromPerson, the toList and the message.

Troubleshooting

To perform maintenance functions, go to your Map Translator Control Panel. Under *Troubleshooting*, you will see the following functions, which you will not need to use every time you change your maps or reconfigure your Map Viewer.

Reopen All Drawings

The software only creates DXF copies of AutoCAD drawing files when the drawing file changes. The software uses checksum values to determine if the drawing changes. If the generation of the DXF files is unsuccessful, it is possible for the software to think the DXF version is correct when it is actually out of date. At least once a month, you should run *Reopen All Drawings* to remove the checksum values for all drawings. During the next translation, the software will reopen and make new DXF files for every drawing, not just the drawings that have different checksums.

Zip Up Logs

If there are problems with translating, we may ask you to zip up your logs and send them to us. Click *Zip Up Logs* to zip up your current logs. This zip file will be placed in the main Partner directory and can be emailed to us for evaluation of a problem.

Zip Up Configuration

There are two reasons use this function: as a back up or to send them to us if we suspect a problem. Click *Zip Up Configuration* to zip up your configuration files, which will be placed in the main Partner directory. By renaming the zip files to a meaningful name, you can have a backup of your configuration Also, if we request, you can email the zipped files to us for evaluation of a problem.

Interpreting the Logs

Here are some of the common error messages that you may see in the translator.

Translation Fails with a *UTFDataFormatException*

This is generally means that a Map Viewer (such as Configure Map Viewing) is running while translation is running. Close all open Map Viewers. You may also try closing all Partner software programs (including the Map Translator Control Panel) and then restart the map translation. If that doesn't work, try rebooting the computer.

Cause: This is due to either database dump or find item files being locked or held open by another program. Windows does not allow files to be deleted while programs are accessing the same files.

Translation Fails with *OutOfMemoryException*

Verify your settings in the *General* tab of Map Translator Configuration. Ensure that the heap size is larger than cache size by at least 50MB. You may want to make the heap size larger. Having a heap size of 400 is especially useful if you are translating TIGER data or ESRI map sets.

Bad Command or Filename Message when Running a Batch File

The problem occurs when you run a batch file or shortcut. A black window opens stating that there is a bad command or filename. Make sure that you are not using Network Neighborhood in Windows Explorer to see the batch file. Windows will not run batch files unless they are drive mapped. This is by far

the most common reason for this error.

If this problem occurs on a local install, run the following program:

Partner/Platform/resources/CalculateAllChecksums.bat.

Then update the computer again.

If the problem persists, try running *Partner/Platform/resources/RemoveReadOnly.bat*. If this works, replace the Platform folder under Partner with a new copy from the Working Install. If *RemoveReadOnly.bat* does not run, then there is a problem with the Windows installation. Check the path setting in the DOS window for corruption.

Chapter 6. Advanced Configuration

Some configuration of the map translation is maintained by editing TOF files. TOF files are text files and can be changed using a text editor such as TextPad, WordPad, or NotePad. The format and syntax of the TOF files are fairly rigid.

Important

Always make a copy of any TOF file before editing.

Data Type and Style Merging and Splitting

The default naming for data types is configured in the file `RoverDatatypeNaming.tof` and for styles in the files `RoverStyleNaming-Line.tof`, `RoverStyleNaming-Point.tof`, and `RoverStyleNaming-Text.tof`. These files are stored in Partner/MapTranslator/config/preferences.

Styles and data types can be renamed by merging or splitting the names. For styles, the splitting is controlled by `RoverStyleSplitting.tof` and merging by `RoverStyleMerging.tof`. For data types, the splitting is controlled by `RoverDatatypeSplitting.tof` and merging by `RoverDatatypeMerging.tof`. These files are stored in Partner/MapTranslator/config/preferences.

To split styles you need a file named `RoverStyleSplitting.tof` that looks like:

```
{
    "line-dxf:PRIMARY" = ["PHASE", "CONDUCTORSIZE"],
    "point-dxf:TRANS" = ["KVA", "PHASE"]
    }
```

The styles mentioned as keys (e.g., "line-dxf:PRIMARY" will then have the fields listed appended, in order, separated by dashes.

To merge styles, you need a file named RoverStyleMerging.tof that looks like:

```
{
    "MajorRoads" = ("line-arc:ROAD_INTERSTATE", "line-arc:ROAD_MAJHWY"),
    "MinorRoads" = ("line-arc:ROAD_HWY",
            "line-arc:ROAD_LOCAL", "line-arc:ROAD_RURAL")
}
```

The styles mentioned as keys (e.g., "MajorRoads") are the new names for any style in the set given as the value. All of those styles will be renamed with the proper prefix (e.g., "text-") added automatically (this prevents renaming a text style as a point style, which would be bad). Splitting happens before merging.

Data type merging and splitting is configured in the same way. Make sure to use data type names, not style names. For example, data type names do not start with line-, point-, or text-.

Configuring Nomenclatures

Nomenclatures are the grid-based naming schemes used by some utilities. The Partner Map Viewer can be configured to use a nomenclature if the naming system is square to the mapping units and has a nam-

ing scheme based on rectangles. Nomenclatures are used for:

- the grid locations shown on the lower pan bar of the Map Viewer.
- the calculations of drawing extents for AutoCAD drawings for regularly gridded systems where the drawing name specifies it's coordinates and extents. Calculating the extents of drawings avoids the problems of having extraneous items in the drawings outside the intended extents.
- adding grid cell values to items in the maps such as consumer numbers to allow merging with external database information.

Nomenclatures are configured using the **nomenclature** property in `Maps/resources/MapControl.tof`.

Nomenclatural issues are complex, unfortunately. Be ready with graph paper and patience to get the configuration working correctly. There always seems to be quite a bit of trial and error in building a nomenclature. You do not need to run the Map Translator to test a nomenclature. Just restart the Map Viewer to see the results of configuration changes.

The primary function of `Nomenclature.tof` is translating between map coordinates and "map names". These *map names* are simply named regions of map space, usually named according to some sort of gridding scheme that might correspond to actual AutoCAD map files or what-have-you.

Our nomenclature engine has the capability of supporting all manner of schemes, but most of our current customers use some sort of hierarchical grid, so that's what we'll focus on in this section.

Start by defining the **width**, **height**, **originX**, and **originY** of the mapped area. These properties are expressed in the mapping units of the drawings. The **originX** and **originY** are points with values equal to the minimum x-coordinate and y-coordinate of the grid. The **width** and **height** are distances from the origin to the far edge of the grid system. With these four properties, you have defined a box that is the exact boundary of the grid system.

The next step to define the number of columns and rows in the grid and the nomenclature of the grid cells. In the first **GridLevel** property, the index = 0 indicates that this is the top level of the grid system. The **columns** and **rows** properties define the number of cells with the boundaries of the grid system. The **start** property gives the number for the upper-left cell. Cells are numbered from left to right, then wrapped to the next row.

The number of columns is important because it controls the wrapping of the cell numbers. You must make sure that you have defined a grid that has exactly the correct number of cells. The key is to have the initial boundary definition and arrangement of cells make the numbering of cells make sense.

The **prefix** property allows you to have a prefix put on the front of the grid cell number for display. This is most common in the lower grids where the cell name includes the upper grid name, a dash or other symbol, then the lower grid name. The Map Viewer will always display the higher level grid cell numbers at the beginning of any lower cell number.

The **numeric** property controls whether to use numbers or letters to name the grid cells. If numeric is false, the grid cells are given sequential letter names starting with letter specified in the **start** property.

The **lowerLeft** property controls the grid cells are numbered starting in the upper right cell or the lower left cell.

The **multiplier** property is an integer that multiplies the grid cell number. This could be used, for example, if the client only used even grid cell numbers.

For any levels below 0, each level assumes that you are subdividing one of the grid cells from the next

higher level grid.

The **nameMapping** property allows you to completely reset the grid cell names on a cell-by-cell basis. The calculated cell name is first and the desired cell name is second.

Example 6.1. A Nomenclature in `Maps/resources/MapControl.tof`

```
#0: new com.PartnerSoft.apps.MapBrowser.MapBrowserConfig {
    "backgroundColor" = "white",
    "birdseyeScales" = #1: [
        -1,
        0,
        0,
        0,
        1,
        2,
        2,
        2
        ],
    "detailZoom" = 5,
    "findItemColorName" = "magenta",
    "foregroundColor" = "black",
    "initialCoord" = #2: new com.PartnerSoft.graphics.Coord3D {
        "x" = 2446100.0,
        "y" = 1500100.0,
        "z" = 0.0
        },
    "initialScale" = 0.0,
    "nomenclature" = #3:
      new com.PartnerSoft.apps.MapBrowser.CustomerMapNomenclature {
        "translator" = #4:
          new com.PartnerSoft.apps.MapBrowser.GridMapTranslator {
            "scaleToGridLevel" = #5: [
                1,
                3,
                3
                ],
            "system" = #6: new com.PartnerSoft.graphics.GridSystem {
                "height" = 270000.0,
                "levels" = #7: [
                    #8: new com.PartnerSoft.graphics.GridLevel {
                        "columns" = 13,
                        "index" = 0,
                        "lowerLeft" = false,
                        "multiplier" = 1,
                        "nameMapping" = #9: {
                            "91" = "",
                            "63" = "851",
[...cut for brevity's sake...]
                            "108" = "",
                            "71" = "612",
                            "116" = "",
                            "21" = "423",
                            "10" = "",
                            "22" = "433",
                            "73" = "822",
                            "98" = "814"
                            },
                        "numeric" = true,
                        "prefix" = "",
```

```
                    "rows" = 9,
                    "start" = 1
                    },
                #10: new com.PartnerSoft.graphics.GridLevel {
                    "columns" = 60,
                    "index" = 1,
                    "lowerLeft" = true,
                    "multiplier" = 1,
                    "nameMapping" = #11: {
                        "3" = "03",
                        "5" = "05",
                        "7" = "07",
                        "2" = "02",
                        "9" = "09",
                        "4" = "04",
                        "8" = "08",
                        "1" = "01",
                        "6" = "06"
                        },
                    "numeric" = true,
                    "prefix" = "-",
                    "rows" = 1,
                    "start" = 1
                    },
                #12: new com.PartnerSoft.graphics.GridLevel {
                    "columns" = 1,
                    "index" = 2,
                    "lowerLeft" = true,
                    "multiplier" = 1,
                    "nameMapping" = #13: {
                        "3" = "03",
                        "5" = "05",
                        "7" = "07",
                        "2" = "02",
                        "9" = "09",
                        "4" = "04",
                        "8" = "08",
                        "1" = "01",
                        "6" = "06"
                        },
                    "numeric" = true,
                    "prefix" = "",
                    "rows" = 60,
                    "start" = 1
                    }
                ],
            "originX" = 2276607.0,
            "originY" = 1390083.0,
            "width" = 390000.0
            }
        }
    },
"numberOfZooms" = 0,
"scales" = #14: [
    1100.0,
    150.0,
    50.0,
    15.0,
    5.0,
    2.0,
    1.0,
    0.3
    ],
"sources" = #15: [
```

```
        "rover"
        ],
    "toggleZoomingAtDetail" = false,
    "viewDataColorName" = "cyan",
    "zoomFactor" = 0
    }
```

It is also possible to have independent nomenclatures as of version Partner version 3.7.1. This is done by using the "subTranslators" keyword. Here is an example:

Example 6.2. Multiple, Independent Nomenclatures example

```
"nomenclature" = #3:
  new com.PartnerSoft.apps.MapBrowser.CustomerMapNomenclature {
    "translator" =
      new com.PartnerSoft.apps.MapBrowser.MapLocationMultiTranslator {
        "subTranslators" = [
            new com.PartnerSoft.apps.MapBrowser.GridMapTranslator {
              "system" = #12: new com.PartnerSoft.graphics.GridSystem {
                  "originY" = 1266670.0,
                  "width" = 160000.0,
                  "height" = 238000.0,
                  "levels" = #13: [
                      #14: new com.PartnerSoft.graphics.GridLevel {
[.. cut out for brevity's sake ...]                                  }
                      ],
                  "originX" = 2046088.0
                  }
              },
            new com.PartnerSoft.apps.MapBrowser.GridMapTranslator {
              "system" = #12: new com.PartnerSoft.graphics.GridSystem {
                  "originY" = 1273727.0,
                  "width" = 176540.0,
                  "height" = 241920.0,
                  "levels" = #13: [
                      #14: new com.PartnerSoft.graphics.GridLevel {
[.. cut out for brevity's sake ...]
                      },
                  "multiplier" = 1
                  }
              ],
            "originX" = 2026067.0
            }
          }
        ]
      }
    }
```

More Translation Options: *MapTranslator.tof*

There are a few options to the translator that are not available in *Specify Data Sources* or *Configure Translator*. For some things, you have to edit files directly. Here are a few of the options. **Note:** Make sure you have a backup before editing these files.

Ignoring Tables in ESRI

You may have some tables in your ESRI-based map system that may be so large that you don't want to use them. In order to ignore tables while translating ESRI, you will need to directly edit *MapTranslator\config\preferences\MapTranslator.tof*, adding an "ignoreTables" list.

ESRI translators should have a section similar to the following, possibly without the ignoreTables section. If ignore tables is already present, just add to the list.

Example 6.3.

```
new com.PartnerSoft.apps.MapTranslator.arcinfo.ArcInfoFrontend {
    "scaleX" = 0.0,
    "offsetY" = 0.0,
    "prescaleX" = 0.0,
    "offsetX" = 0.0,
    "rotationFields" = (
        "rotationfield",
        "textrotation",
        "rotation",
        "rotationangle",
        "text_angle",
        "angle",
        "symbolrotation",
        "textangle",
        "sym_angle"
        ),
    "ignoreTables" = (
        "Capacitor",
        "Electric_Net_Junctions",
        "Fuse",
        "Recloser",
        "Regulator",
        "Sectionalizer",
        "Switch",
        "Transformer"
        ),
    "prescaleY" = 0.0,
    "preoffsetY" = 0.0,
```

If you don't already have an ignore tables, you can add the following into MapTranslator.tof:

```
    "ignoreTables" = (
        ),
```

Then add the tables to ignore to the list, one per line, enclosed with quotes with a comma after each except the last.

ESRI Rotation Fields

You may notice in the above example that you can also set the rotation fields that the translator will use to set rotation on points and text. If the Map Translator is not rotating text or symbols for some reason in your map set, you may be able to fix in in *MapTranslator.tof*. First, be sure that rotation is actually set in

the ESRI data. Next, determine the database field that contains the rotation for text or points. Add the rotation field, as shown above, to the rotationFields list.

Adding Pictures and Other Raster Images

The Map Viewer supports presenting pictures associated with items on the maps. Pictures can include photographs of equipment, plots of wiring diagrams, or scanned copies of paper documents. Each picture must be associated with an entity on the map. When the user views the data for the map entity, one of the data values is a button for viewing the image. Clicking the button brings up a raster image viewer.

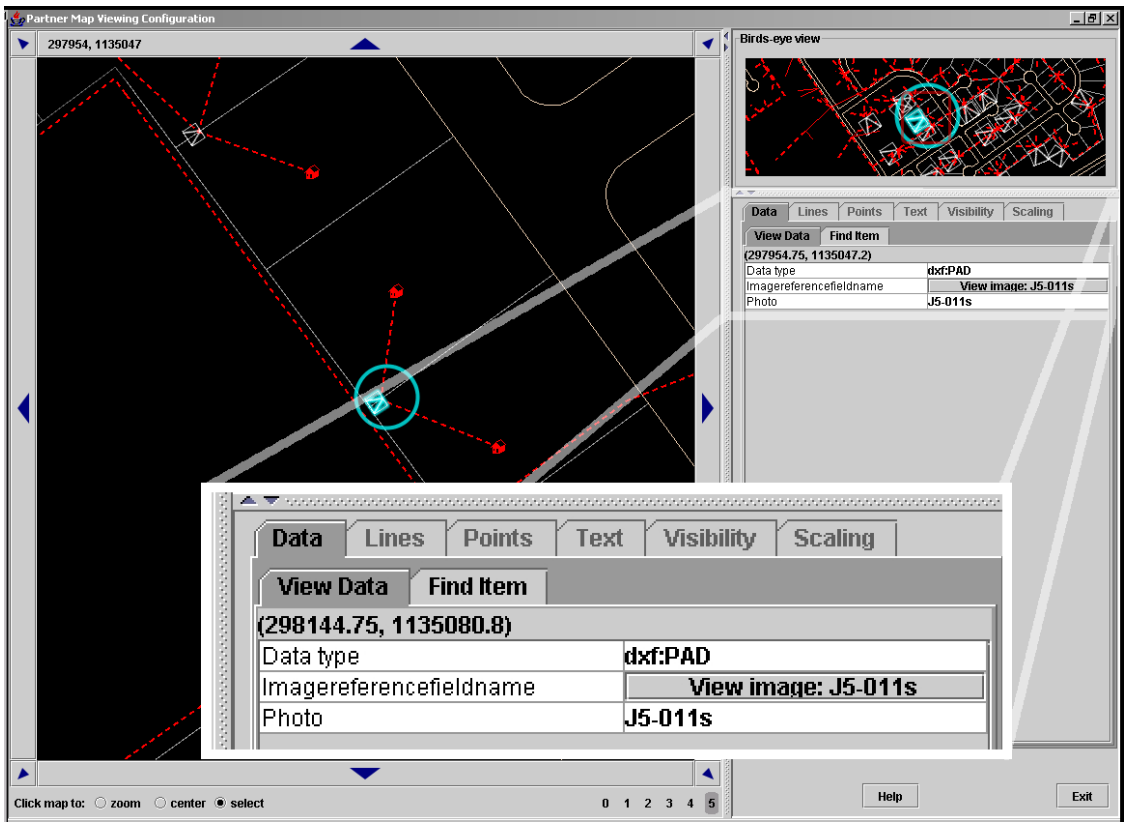


Figure 6.1. The Map Viewer with highlighted padmount transformer and button added for linked picture.

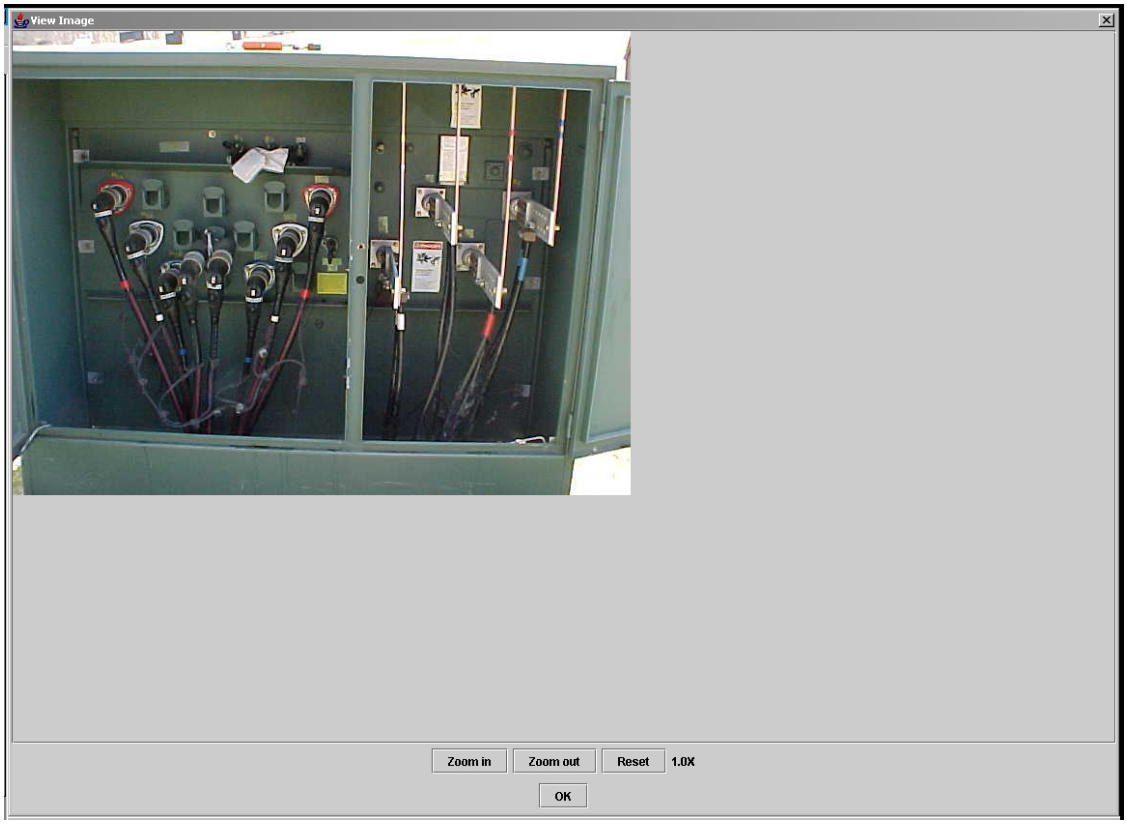


Figure 6.2. Padmount transformer picture.

Configuration

First things first -- An external file (CSV or other) must establish a link between the map location and the photo name. If this file does not exist, it must be created.

For example, the photos can be linked to the padmount transformer by the unique transformer number. The external file should then contain a field for the transformer number (e.g. "TRANS_NUM") and a field for the associated photo file (e.g. "PHOTO" -- The names given to the fields are your choice, but keep them simple and descriptive of their content). In Example 6.4, the values under the PHOTO field are simply the names of the photo files with the file extension excluded.

Example 6.4. Sample external CSV file linking map location and photo name.

```
TRANS_NUM, PHOTO
1456874, J5-011s
1456875, M11-017f
1456902, B1-019s
...    , ...
1457345, B3-021s
```

This file, when completed, must be first merged into the translator and then added to the map items themselves. In this example, the field "TRANS_NUM" will be your index field, and your merge field will be a similar field in the padmount datatypes (see Merging External Data for instruction on this step).

The photos or raster images need to then be placed in a folder within the *Partner/Maps/resouces* folder such as *Partner/Maps/resources/photos*. Furthermore, the location of this folder needs to be defined as part of an imageLibraries statement within the file *Partner/Maps/resources/rover/MapDataSource.tof*, as is done in example 6.5 below:

Example 6.5. Editing the MapDataSource file.

```
"imageLibraries" = {
    "PadPhotos" = "../photos",
    "Rover Icons" = [
        "../icons/standard",
        "../icons/custom"
        ]
    }
```

In this example, the ImageLibraries statement 'PadPhotos' is defined for all images stored in the *Partner/Maps/resources/photos* folder. Next, a Python transform such as the one in Example 6.6 must be added to the map item ("dxf:PAD" in this case):

Example 6.6. Python transform text.

```
from com.PartnerSoft.gui import ImageReference;

imageName = original.getString('PHOTO');
reference = ImageReference('PadPhotos', imageName);
original.setObject('imageReferenceFieldName', reference);
```

This transform changes the merged name field ('PHOTO', from Example 6.4) into the image reference ('PadPhotos') created above in *MapDataSource.tof*. It also adds a link button with the name of the image to the new field 'imageReferenceFieldName'. To see this button (Figure 6.1), enable this field in the translator and translate . For assistance on creating Python transforms, click here.

Adding Construction Job Information

If you are using the Field Designer software, you can add icons for construction jobs with the information from the job header fields to the Map Viewer. You need to add a new data record source to *Partner/MapTranslator/config/preferences/MapTranslator.tof*. Below is an example of that data record source.

```
new com.PartnerSoft.apps.MapTranslator.data.DataRecordSourceFrontend {
    "scaleX" = 1.0,
    "sources" = [
        new com.PartnerSoft.io.sql.SQLDataRecordSource {
            "query" = "Select
```

```
            *
            from jobs",
          "showProgress" = false,
          "transformScript" = "
x=original.getInt('centroidX');
y=original.getInt('centroidY');
original.setString('databaseName','partner');
original.setString('tableName','jobs');
original.setString('x',str(x));
original.setString('y',str(y));
          ",
          "verbose" = false,
          "database" = new com.PartnerSoft.io.sql.DatabaseConnectionConfig {
              "databaseServer" = "localhost",
              "databaseUser" = "root",
              "databaseName" = "Partner",
              "databaseType" = "MySQL",
              "databasePassword" = "",
              "databasePath" = ""
              }
          }
      ],
  "offsetY" = 0.0,
  "verbose" = true,
  "scaleY" = 1.0,
  "offsetX" = 0.0,
  "name" = "dat"
  }
```

If the Map Translator is on the same computer as the System Hub, you can use the above example. If the Hub is on a different computer, then you need to specify the databaseServer for the Hub computer.

After the first translation, new styles, and data types will be available under Configure Translator.

Symbols

The Map Viewer uses symbols for annotating sketches and for presenting mapping features.

Creating a Symbol File

The Map Viewer can use GIF, PNG, JPG, or JPEG files for symbols. There are many programs that create these files; for example, Paint Shop Pro. Using version 4.0 of this program, under File, Preferences, File Format Preferences, GIF select Set the transparency to the background color (see Appendix A). This allows the GIF to have a transparent background, which is the appropriate setting for all symbols except *Hide Blocks*. The Field Designer generally has a black background so you can also set the color black to transparent. During printing of the sketch, the Field Designer and Map Viewer are generally configured to convert the white of symbols to black so instead of white symbols on a black background, you have black symbols on a white background.

We recommend saving your images as GIF or PNG files.

Special Characteristics

A minus ('-') in front of the name of an redline symbol in the *AnnotationSymbols.tof* file specifies that the symbol should be written prior to writing other redline and job entities. This has the result of writing the base map, the "minus" redline symbols, then the rest of the sketch. This feature is used for *Hide blocks* so that they do not hide redline for the job.

Management of Redline Symbols

The redline symbol files are stored in the */Shared/resources/icons/annotation* subdirectory. New symbol files can be added to this subdirectory.

The list of redline symbols available to the Map Viewer and Operations Manager is controlled by the file */MapViewer/Config/Partner/AnnotationSymbols.tof*. The symbols appear on the list in the Map Viewer in the same order as in the TOF file. To add a redline symbol, create the symbol file you want and name with the text you want to appear in the selection list. Add the name to the list of available symbols in the *AnnotationSymbols.TOF file*. To remove a symbol from a program, remove the name from the *AnnotationSymbols.tof* file. It is permissible to leave the symbol file in the */ Shared/Resources/Icons/Annotation* subdirectory if desired.

Management of Mapping Symbols

The mapping symbol files are stored in the */Maps/Resources/Icons/custom* and */ Maps/Resources/Icons/standard* subdirectories. The standard directory should contain only symbols provided by Partner as part of our standard symbol set. The custom directory contains symbols specifically generated for the client. If the same symbol name is in both directories, the software uses the symbol in the custom directory. If you change or add symbols, make sure you are working on symbol files in the custom directory. It's fine to copy files from the standard directory into custom and then modify the files.

You control the use of symbols using the Map Viewing Configuration option of the Map Translator as described in that section of this manual.

Management of Company Logo

The company logo is stored as *shared/resources/icons/ClientLogo.png*. It can also be a gif file (*Client-Logo.gif*). If this file is present, then this logo is displayed instead of the Partner Software logo over the birdseye at zoom 0 of the Map Viewer. The image should be about 150 pixels high.

Changing the Color of Your Find Item Arrow

If the default color for your *Find Item* arrow needs to be changed, follow these steps:

- Go to *Partner/Maps/resources/icons/standard* and move the file *find-item.png* to *Partner/ Maps/resources/icons/custom*.
- Open the *find-item.png* file in GIMP or another graphic program and change the color. Save it to the *custom* folder.

Chapter 7. Importing ESRI Shapefiles from the Partner Field Designer into AutoCAD

Components

This document assumes the reader has a working knowledge of the Partner Field Designer, as well as AutoDesk's AutoCAD system (Map version 3, 4, or 5).

The following components are included in this process:

- an entry in *FieldDesigner\config\partner\reports\index.tof* to turn on the export of shapefiles;
- *ConfigManager\config\customer\UnitInfo* with CSV files for the units that are going to be imported into AutoCAD (e.g., we will show a *poles.csv* file to export poles from the Field Designer to the *POLEJC* AutoCAD layer and the *POLE* AutoCAD block;
- an entry in *FieldDesigner\config\partner\reports\Shapefile.tof*;
- the shapefile lisp routine for importing shape files into AutoCAD.

Turning on Shapefile Export in the Field Designer

Edit *FieldDesigner\config\partner\reports\index.tof,* and add a line *"Shapefile"* to the file, shown below in the example.

```
[
    "AccountSummary",
    "CADPMapExport",
    "Shapefile",
    "StakingSummary"
    ]
```

Creating the Data Items for Export

In order to export shapefiles that will load the correct blocks and into the correct layers in AutoCAD, more configuration must be done in the Field Designer. A comma-separated file (CSV) file needs to be created for each of the units that you want to export. It is easiest to create a separate CSV file for each unit type you are exporting.

The CSV file must contain, at a minimum, a header with the fields *unit, includeInUnitPoints*, and *unitStatus*. Fields named *block* and *layer* would normally also be included. For each line of the CSV file, the unit must named as it is named in the Partner System. The layer and block fields will contain the respective names of the AutoCAD layer and block that unit will be loaded into. The value for *includeInUnitPoints* on each line should be *y*, and *unitStatus* should be *`unitAtALocation.status`*(The back ticks are required).

When you save the file, be sure to save it as a CSV file.

Here is an example of creating a CSV file using a spreadsheet program, such as Microsoft Excel.

Example 7.1. Creating a CSV File in a Spreadsheet

	A	B	C	D	E	F
1	unit	layer	height	includeInUnitPoints	unitStatus	block
2	20' AL. POLE	POLE	20	y	`unitAtALocation.status`	POLEJC
3	25' AL. POLE	POLE	25	y	`unitAtALocation.status`	POLEJC
4	25' POLE	POLE	25	y	`unitAtALocation.status`	POLEJC
5	30' POLE	POLE	30	y	`unitAtALocation.status`	POLEJC
6	35' POLE	POLE	35	y	`unitAtALocation.status`	POLEJC
7	40' POLE	POLE	40	y	`unitAtALocation.status`	POLEJC
8	45' POLE	POLE	45	y	`unitAtALocation.status`	POLEJC
9	50' POLE	POLE	50	y	`unitAtALocation.status`	POLEJC
10	55' POLE	POLE	55	y	`unitAtALocation.status`	POLEJC
11	60' POLE	POLE	60	y	`unitAtALocation.status`	POLEJC
12	65' POLE	POLE	65	y	`unitAtALocation.status`	POLEJC
13	70' POLE	POLE	70	y	`unitAtALocation.status`	POLEJC
14	75' POLE	POLE	75	y	`unitAtALocation.status`	POLEJC
15	80' POLE	POLE	80	y	`unitAtALocation.status`	POLEJC
16	85' POLE	POLE	85	y	`unitAtALocation.status`	POLEJC

The following is the same CSV file in a text editor, such as Textpad.

Example 7.2. Creating a CSV file in a Text Editing Program

```
unit,layer,height,includeInUnitPoints,unitStatus,block
20' AL. POLE,POLE,20,y,`unitAtALocation.status`,POLEJC
25' AL. POLE,POLE,25,y,`unitAtALocation.status`,POLEJC
25' POLE,POLE,25,y,`unitAtALocation.status`,POLEJC
30' POLE,POLE,30,y,`unitAtALocation.status`,POLEJC
35' POLE,POLE,35,y,`unitAtALocation.status`,POLEJC
40' POLE,POLE,40,y,`unitAtALocation.status`,POLEJC
45' POLE,POLE,45,y,`unitAtALocation.status`,POLEJC
50' POLE,POLE,50,y,`unitAtALocation.status`,POLEJC
55' POLE,POLE,55,y,`unitAtALocation.status`,POLEJC
60' POLE,POLE,60,y,`unitAtALocation.status`,POLEJC
65' POLE,POLE,65,y,`unitAtALocation.status`,POLEJC
70' POLE,POLE,70,y,`unitAtALocation.status`,POLEJC
75' POLE,POLE,75,y,`unitAtALocation.status`,POLEJC
80' POLE,POLE,80,y,`unitAtALocation.status`,POLEJC
85' POLE,POLE,85,y,`unitAtALocation.status`,POLEJC
```

The decision needs to be made as to what shapefiles need to be exported. Lines and points must be exported from separate shapefiles. Most of the values used when importing into AutoCAD can be found in

the CSV files made above. See the following example.

Example 7.3. Setting for Field Designer to Export a Shapefile Called *Unitpoints*

Editing *FieldDesigner\config\partner\reports\Shapefile.tof*:

```
new com.PartnerSoft.apps.Staker.interfaces.shapeFile.ShapeFileJobTheme {

  "filter" = [
    "com.PartnerSoft.apps.Staker.domain.UnitAtALocation:unitAtALocation.un
    ],

  "dataSettings" = [
      new com.PartnerSoft.io.formats.DBFFieldDescriptor {
          "fieldName" = "LAYER",
          "fieldValue" = "unitAtALocation.unit.info.layer",
          "fieldType" = 'C',
          "fieldLength" = 20
          },
      new com.PartnerSoft.io.formats.DBFFieldDescriptor {
          "fieldName" = "BLOCK",
          "fieldValue" = "unitAtALocation.unit.info.block",
          "fieldType" = 'C',
          "fieldLength" = 20
          },
      new com.PartnerSoft.io.formats.DBFFieldDescriptor {
          "fieldName" = "HEIGHT",
          "fieldValue" = "unitAtALocation.unit.info.height",
          "fieldType" = 'C',
          "fieldLength" = 20
          },
      new com.PartnerSoft.io.formats.DBFFieldDescriptor {
          "fieldName" = "STATUS",
          "fieldValue" = "unitAtALocation.unit.info.unitStatus",
          "fieldType" = 'C',
          "fieldLength" = 20
          }
      ],

  "name" = "UnitPoints",

  "trueLocation" = false,
  "pixelToFeet" = 3.0,

  // points
  "shapeType" = 1
  }
```

Importing Shapefiles into AutoCAD

You can use the **mapimport** (AutoCAD 14) or **mapimport4** (AutoCAD 2000) to import the shapefiles manually. The rest of this section describes the automated update scripts.

Configuring the AutoLSP Import Program

The shapefiles can be imported using an AutoLSP script. Create a script like that listed below to load a particular type of shapefile. The name of the AutoLSP script is variable, but for this example we are using ShapeFileImport.lsp.

Example 7.4. Listing of *ShapeFileImport.lsp*

```
;;;copyright 2001 Partner Software;;
;;;-------------------------------------------------------------------;
(setq workingDir "C:/Partner/FieldDesigner/reports/jobs/")
(setq importCommand "mapimport")
(setq settingsFile  "shapefileimport.dat")

;;;-----------------------------------------------------------;
;;;   Function: C:ShapeFileImport                             ;
;;;-----------------------------------------------------------;
;;; Description: load shape files into AutoCAD drawings       ;
;;;     to update Partner maps.                               ;
;;;-----------------------------------------------------------;
(defun C:ShapeFileImport (/ startCmd startFile jobNumber jobFile)
  ;; Get environment variables and reset them to more useful values
  (sfi:envSetup)
  ;; Ask the user for input to decide which job will be updated
  (setq jobNumber (getstring "\Enter the job number: "))

  ;; Load the shapefile

  (sfi:importJob)
  ;; Restore saved environment variables
  (sfi:envRestore)

  )

;;;-----------------------------------------------------------;
;;; Function: sfi:envSetup                                    ;
;;;-----------------------------------------------------------;
;;; Description: reset environment variables                  ;
;;;-----------------------------------------------------------;
(defun sfi:envSetup ()
  ;;get some of the current environment variables and remember them
  (setq startCmd (getvar "cmddia"))
  (setq startFile (getvar "filedia"))
  ;;set some of the environment varibles to disable dialogue boxes
  (setvar "cmddia" 0)
  (setvar "filedia" 0)
  )

;;;-----------------------------------------------------------;
;;; Function: sfi:importJob                                   ;
;;;-----------------------------------------------------------;
;;; Load the job into the drawing. Duplicate the two lines of  ;
;;; of the command for as many shape files that need to be     ;
;;; loaded                                                    ;
;;;-----------------------------------------------------------;
(defun sfi:importJob ()
    ;; Set the variable jobFile to the path of the job
  (setq jobFile (strcat workingDir jobNumber"/UnitPoints.shp"))
  (command importCommand "shp" jobFile "Y" settingsFile)
```

```
    )

  ;;;-------------------------------------------------------------;
  ;;; Function: sfi:envRestore                                    ;
  ;;;-------------------------------------------------------------;
  ;;; Description: set the environment variables back to the;
  ;;;       original values                                       ;
  ;;;-------------------------------------------------------------;
  (defun sfi:envRestore ()
    ;;reset the environment variables to their original values
    (setvar "cmddia" startCmd)
    (setvar "filedia" startFile)
    )
```

The line "(defun C:ShapeFileImport (/ startCmd startFile jobNumber jobFile)" contains the specification for the word typed on the command line to start the script. In this case the command is "ShapeFileImport".

The line "(setq workingDir "U:/Partner/WorkingInstall/reports/jobs/")" specifies the path to the jobs directory on the Working Install. It must match the drive mapping on the computer used for mapping. Each of the folders within the jobs folder will be a job number. Each of these folders will contain the shape files if the system has been set up properly.

The line "(setq importCommand "mapimport")" specifies that the mapimport command should be used. For AutoCAD 2000 and later change this to "mapimport4".

The line "(setq settingsFile "shapefileimport.dat")" specifies the path and name of the saved settings for importing the shapefile. The next sections describes creating this settings file.

For each additional shapefile that needs to be loaded, copy and paste the item:

```
(setq jobFile (strcat workingDir jobNumber"/UnitPoints.shp"))
(command importCommand "shp" jobFile "Y" settingsFile)
```

Change "UnitPoints" in the above to the name of the shapefile you want to load. For example, to load another shapefile named ConstructionUnits.shp, make the load file section look like this:

```
;;;-------------------------------------------------------------;
;;; Function: sfi:importJob                                     ;
;;;-------------------------------------------------------------;
;;; Load the job into the drawing. Duplicate the two lines of  ;
;;; of the command for as many shape files that need to be     ;
;;; loaded                                                      ;
;;;-------------------------------------------------------------;
(defun sfi:importJob ()
  ;; Set the variable jobFile to the path of the job
    (setq jobFile (strcat workingDir jobNumber"/UnitPoints.shp"))
    (command importCommand "shp" jobFile "Y" settingsFile)

    (setq jobFile (strcat workingDir jobNumber"/ConstructionUnits.shp"))
    (command importCommand "shp" jobFile "Y" settingsFile)

  )
```

Creating a Saved Loading Configuration

Start AutoCAD. Turn CMDDIA and FILEDIA to 1. Type mapimport or mapimport4. Specify a shape file to import and click proceed. Choose the settings you want to use for importing that kind of shapefile. The settings include saving data to object data tables, naming blocks, and specifying layers. Save the settings file to disk when you get it working. This is the settings file specified in the AutoLSP script.

Loading the LSP Program into AutoCAD

Start AutoCAD. Choose Tools, AutoLSP, Load. Browse to the location of the LSP program.

Running the LSP Routine

Type the command name from the LSP file on the command line. In the example, the command is ShapeFileImport.

Chapter 8. Python Scripts

Python is a powerful scripting language that has been around for many years. The Partner System uses a modified subset of this language called Jython to actually run Python scripts. From the perspective of the end user, the language is Python. There are a variety of books on Python at your local bookstore, such as *Learning Python*, by Lutz and Ascher (O'Reilly), which will help with getting started with Python. The web site *http://www.python.org* has full Python documentation. Though many elements within the Partner system are scriptable through the use of Python, this section will only cover using Python with the Partner Map Translator.

How Python and Partner Interact

Partner native functions can be used from within Python scripts. Primarily, one should use the set of commands for reading and writing data between Python to Partner. Whenever using a Partner function from within Python (such as original.getString('')), each line must end in a semi-colon. It is fine to end all lines with semi-colons, though Python itself does not require this.

In the following examples, the field name *variable2* would be a field similar to those displayed in the Map Translator—*Data Types* screen.

Get floating point, integers, strings, or objects using the following:

- Read variable2 from Partner into the Python as the floating point number with a name of variable1. Variable2 must be a valid floating point number.

  ```
  variable1 = original.getFloat('variable2');
  ```

- Read variable2 from Partner into the Python as the integer variable1. Variable2 must be a valid integer.

  ```
  variable1 = original.getInt('variable2');
  ```

- Read variable2 from Partner into the Python as the long-integer variable1. Variable2 must be a valid integer.

  ```
  variable1 = original.getLong('variable2');
  ```

- Read the field variable2 from Partner into the Python as a string called variable1.

  ```
  variable1 = original.getString('variable2');
  ```

- Create a Python object variable1 from an object from a Partner object, variable2.

  ```
  variable1 = original.getObject('variable2');
  ```

Set floating point, integer, or string values using the following:

- Write variable1 from a Python script to Partner as a floating point number variable2. Variable1 must be a valid floating point number.

```
original.setFloat('variable2',variable1);
```

- Write the field variable1 from Python to Partner as an integer named variable2. Variable1 must be a valid integer.

  ```
  original.setInt('variable2',variable1)
  ```

- Write the field variable1 from Python to Partner as a long-integer named variable2. Variable1 must be a valid integer.

  ```
  original.setLong('variable2',variable1)
  ```

- Write variable1 from a Python script to Partner as the string variable2.

  ```
  original.setString('variable2',variable1);
  ```

- ```
  variable1 = original.addSequenceNumber('variable2', 'sequenceNumber', 'GRID Moni
  ```

Other variable manipulations:

- Make a copy of variable1 as variable2 in the original data.

  ```
  original.copy('variable1', 'variable2');
  ```

- Rename variable1 as variable2 in the original data.

  ```
  original.move('variable1', 'variable2');
  ```

- Delete variable1.

  ```
  original.remove('variable1', 'variable2');
  ```

In order to appear in the Partner logs, you will need to use the following syntax:

```
log.enter('This is a normal log entry');

log.warning('This warning log entry is yellow in the Partner logs.');

log.error('This error log entry is displayed in red in the Partner logs.');

log.enter('Display the values of variables ' + Variable + '
            using the + sign as a separator.');
```
```

Using Python

Here are a few commonly used functions.

String Manipulations

- Create a new variable name1 by concatenating name2 and name3 with a comma space in between.

```
name1 = name2 + ', ' + name3;
```

- Create string variable name1 from numeric variable name2.

```
name1 = str(name2);
```

- Pad the string variable name1 to 11 characters by adding leading zeros. The indent is important in Python because it indicates the lines within the loop.

```
while len(name1) < 11:
 name1 = '0' + name1;
```

- Create a string variable name1 by converting the string variable name2 to upper case. You must use the import string statement earlier in the Python script to import the string library. String functions include:

```
import string
name1 = string.upper(name2);
```

- Change the first letter of the first word to upper case and all other letters to lower case.

```
string.capitalize(name1);
```

- Pad the front and back of name1 with spaces to make it the length specified in width.

```
string.center(name1,width);
```

- Convert all letters in name1 to lower case.

```
string.lower(name1);
```

- Remove leading spaces from name1.

```
string.lstrip(name1);
```

- Remove trailing spaces from name1.

```
string.rstrip(name1);
```

- removes leading and trailing spaces from name1.

```
string.strip(name1);
```

- Create a string variable name1 by taking the first 6 characters of name2

```
name1 = name2[0:5];
```

- Create a string variable name1 by removing the first 2 characters from name2.

```
name1 = name2[2:];
```

- Create a string variable name1 by retaining the first 3 characters from name2.

```
name1 = name2[:2];
```

- Another way to pad a variable is with string.zfill. The following creates a of value '00000012345'.

```
string.zfill('12345', 11);
```

- Zfill understands periods and negative signs as part of the padding. zfill will not truncate numbers that are too long. The following will give a value of is '-0000003.14'.

```
string.zfill('-3.14', 11);
```

Error Handling: Try—Except

If errors happen during data manipulation with python, it can crash the translator. To avoid this, you should enclose statements within a try—except routine, as in the following example that creates a number padded to 16 characters.

```
import string;
account  = original.getString('Customer Number');
paddedaccount=''
try:
paddedaccount = string.zfill(account,16);
original.setString('CustomerMergeNumber',paddedaccount);
except:
log.warning('Error Padding Account');
```

In the above example, a Customer Number such as 2354256 would create a field in the Map Translator Data Types called CustomerMergeNumber, and for this consumer, would have the value of 0000000002354256. If there is a problem creating the CustomerMergeNumber, the Map Translator will continue to translate.

In order to catch the Java line number of the error, you will need a more elaborate phrase, as follows.

```
from com.PartnerSoft.util.SystemLog

try:
  [do something]
except Exception, something:
    SystemLog.singleton().error(something)
```

Examples

Formatting a Telephone Number

Scenario: a field contains 10 digit numbers that are phone numbers. Users want the number formatted so that it is in standard telephone format of (xxx) xxx-xxxx. The following script shows how to do this if you assume that formatting should be done by starting at the right part of the string and traveling left.

```
phone = original.getString('Phone')
FormattedPhone=phone
try:
  if len(phone)<8:
    FormattedPhone=phone
  else:
    FormattedPhone=phone[:-10]+'('+phone[-10:-7]+')'+phone[-7:-4]+'-'+phone[-4:]
except: pass
original.setString('FormattedPhone',FormattedPhone)
```

This will create a field called FormattedPhone in the form (xxx) xxx-xxxx. Note that for phone numbers greater than 10, the script will include the extra numbers/characters at the beginning. The assumption in this script is that there may be seven digit numbers, and there may be numbers that are more than ten digits. By counting from the right, phone numbers that are shorter than expected will still have the best fit formatting applied to them. The string

```
FormattedPhone=phone[:-10]+'('+phone[-10:-7]+')'+phone[-7:-4]+'-'+phone[-4:];
```

works like this:

```
[Characters more than 10 characters from right], [a left parenthesis],
[characters 7-10 from right], [a right parenthesis],
[characters 4-7 from right], [a dash],
[characters 4 characters from right]
```

If the script encounters an error, Formattedphone will be set to the same value as phone.

Splitting Graphical Elements

When splitting graphical element using a a data field and RoverStyleSplitting, it is often advisable to create a new data field with the expected values for splitting. If a value that is not expected is encountered, the split field will have the value 'other'. This is also a good way to create the appropriate split values in the case that one wants the graphical elements named differently.

```
import string;
SecType=original.getString('TYPE');
SplitField='other';

if SecType[-6:]=='SECONDARY-OH':
SplitField='SEC-OH';
```

```
if SecType[-6:]=='SECONDARY-UG':
SplitField='SEC-UG';
if SecType[-7:]=='PRIMARY-UG':
SplitField='PRIM-UG';
original.setString('SplitField',SplitField);
```

Regular Expression

Quite often databases will contain the building number with street address as a single field. However, in the find item, it is quite useful to be able to search by street name, then the building number, such as Main St., 123 instead of 123 Main St. The following script will get the service_address variable, split the addresses based on the spaces, and test the first 'word' of the address for a numeric value. If that fails, it will try to split the first "word" of the address based on extensions that may be appended to the building number, for example: 123-1/2 Main street.

If either of the above two cases succeed, the script will return a variable, AddressStreetFirst, to the translator such that an original value of 123 Main St. will become Main St., 123. There are other ways to accomplish this what this script does, but using regular expressions is probably the shortest way to do this substitution. Of course, regular expressions are very intimidating to new users mostly due terse nature. A programmer once commented that "Some people, when confronted with a problem, think "I know, I'll use regular expressions." Now they have two problems."

```
ServiceAddress=original.getString('service_address');

AddressStreetFirst=ServiceAddress

import string,re;

try:
# Look for address where the first word begins with at least one
# number, -, or /., followed by a space, and optional numbers,
# A,B,C,D. Move all the preceeding stuff after everything else in
# the line. Works for:
# 123 Main St, 123-1/2 Main St, 123 A Main St, 123 1/2 Main St.
AddressStreetFirst=re.sub('(^[0-9-/]+ ?[1-2/ABCD]* )(.*)','\\2, \\1',
  ServiceAddress,10)
except:
  pass #If all else fails, leave new address field equal to the old
    # address field.

original.setString('AddressStreetFirst',AddressStreetFirst);
```

Converting Multiple Merge Fields to Individual Data Fields

If you do a multiple merge during translation, you will create a new field containing the multiple merge results, e.g., reading a file of attachments and assemblies and merging them to poles. The following example makes individual fields of the results of reading an attachments file.

```
attachmentArray = original.getObject('ATTACHMENTS');
notAssembly = 0;
notAttachment = 0;
try:
```

```
for i in range(attachmentArray.size()):
    currentAttachment = attachmentArray.get(i);
    type = currentAttachment.get('Attachment TYPE');
    note = currentAttachment.get('Note');
    found = 0;
    if type == 'Assembly':
        iString = str(i+1-notAssembly);
        original.setString('Assembly' + iString, note);
        notAttachment = notAttachment + 1;
        found = 1;
    if type == 'Attachment':
        iString = str(i+1-notAttachment);
        original.setString('Attachment' + iString, note);
        notAssembly = notAssembly + 1;
        found = 1;
    if type == 'GPS_LIGHT':
        original.setString('GPS_LIGHT', note);
        notAssembly = notAssembly + 1;
        notAttachment = notAttachment + 1;
        found = 1;
    if found == 0:
        j = j + 1;
        original.setString('otherAttachment', type);
except: pass
```

In the above example, the fields created are Assembly1, Assembly2, ... AssemblyN; and Attachment1, Attachment2, ... AttachmentN as string variables with the value of the Note field from the original data. Also created are the fields GPS_LIGHT and otherAttachment. This entire section was placed in a try structure for the cases with no values in ATTACHMENTS.

Counting the Number of Objects in a Multiple Merge

The following is an example of counting the number of objects in a multiple merge. The field person is the multiple merge field.

```
persons = original.getObject('person');
NumberOfConsumers = persons.size();
original.setInt('NumberOfConsumers',NumberOfConsumers);
```

In the above example, the field person is the multiple merge field. A new field called NumberOfConsumers is created on the original object.

Building a Python Transform

If you wish to add external data to your map configuration, the external data must have an index field to link up with a merge field in the map data. If the values of the records in the index field are different from those records in the merge field (i.e., are arranged differently, have extra characters, etc.), then a Python transform is necessary. See discussion and an example here.

Appendix A. Colors

You can specify a color by name (e.g., "black") or you can specify it using its hexadecimal RGB code a la Netscape (e.g., "#ff0000", which is red). These coded colors are useable anywhere a named color is used. Most image programs support display of color hex codes. You can use either a 6-digit RGB code (e.g., "#ff00ff" = purple) or you can use an 8-digit RGBA code with alpha transparency (e.g., "#ff00ff80" is a 50% transparent purple). Alpha of 00 is clear, while FF is fully opaque. In between are the more useful values—e.g., 0x20 is 32, 0x40 is 64, 0x80 is 128, etc.

You can also specify color codes using comma-separated values—e.g., "(255, 0, 0)" is red and "(255, 0, 255, 128)" is half-opaque purple.

You can also specify color codes using AutoCAD color codes. The format is "x-acad-#" where "#" is the AutoCAD color number. For example, "x-acad-7" is white. If all this confuses you, stick with the named colors.

These colors may be used wherever colors are specified by name in Partner configuration.

- black
- blue
- cyan
- darkGray
- gray
- green
- lightGray
- magenta
- orange
- pink
- red
- white
- yellow

Appendix B. XMLGenerator—A Data Dumper for GE Smallworld

XMLGenerator is a lightweight application that exports records from a ds_collection to a text file, optionally where a specified geometry field is on or within the current view bounds. The output format is a form of XML encapsulating attributes, geometries, and joins. It is not GML compliant, however, being terser in content.

Supported physical and derived field types include User Defined Scalar, String, Integer, Float, Boolean, and Date categories. Supported geometry field types include area, chain, point, text, simple_area, simple_chain, and simple_point. Join fields, including text joins, are supported.

Load XMLGenerator into a Smallworld session at the Magik prompt as follows:

```
load_file("d:\xmlgenerator.magik")
```

XMLGenerator requires a graphics_system as a parameter for initialization. Typically, in Smallworld 3.3 professional activity session, this is obtained as follows:

```
g << gis_program_manager.applications[:professional]
```

In earlier versions of Smallworld, you would normally use the following:

```
g << gis_program_manager.applications[:gis]
```

Using the XMLGenerator GUI

XMLGenerator has an optional GUI that allows the user to select specific tables to export. There is a graphics_system method to launch the GUI, which may be assigned to a menu entry:

```
g.launch_xmlgenerator()
```

The GUI appears as follows:

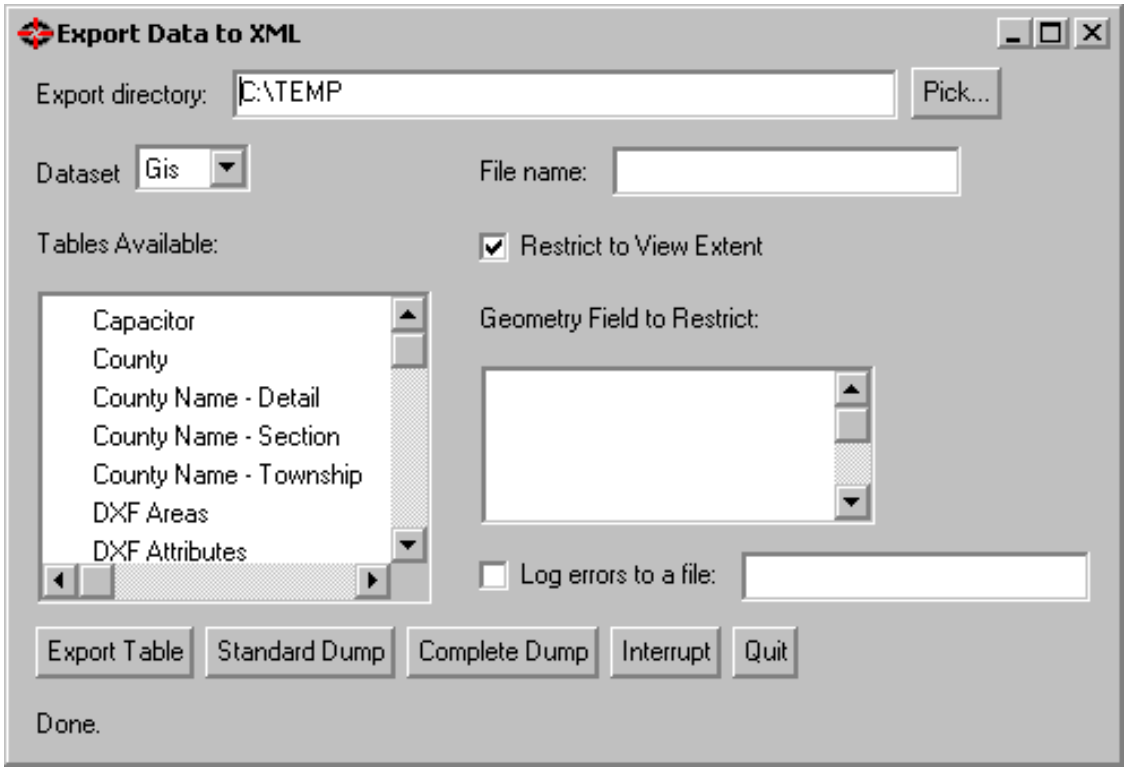


Figure B.1. XMLGenerator GUI

Select the desired partition using the *Dataset* choice item. Tables available for export will be listed below. When a table is selected, a list of available geometry fields is displayed. When the *Restrict to View Extent* option is checked, the selected geometry field is used to restrict the export to the bounds of the current view. The *Export Table* button performs the individual table export.

The *Standard Dump* and *Complete Dump* buttons require no selection of tables or geometry fields or specification of filenames. The *standard* dump exports all tables that are currently visible within the view; file names are determined by the partition and table names, and records are restricted to the view extents using the first geometry field encountered having the highest ranking in the following hierarchy:

```
area  <--- highest
chain
point
simple_area
simple_chain
simple_point
text  <--- lowest
```

The *Complete Dump* exports all records in all user tables.

By default, export errors are ignored. When the *Log errors to a file* option is checked, any errors are recorded to the specified file in the export directory. The *Interrupt* button halts the export process, and the *Quit* button closes the GUI.

Using XMLGenerator at the Magik Prompt

Dumps may also be performed at the Magik prompt without invoking the GUI. This allows automation of the process. XMLGenerator is initialized as follows:

```
x << xmlgenerator.new(g)
```

where G is the graphics_system obtained as described earlier. An alternate methods that attempts to find the appropriate graphics system is:

```
x << xmlgenerator.new_for_session()
```

Tables available in the current session may be listed to a file:

```
x.list_tables(a_filename)
```

Each line of the output file is of the format:

```
A_PARTITION_NAME,A_TABLE_NAME
```

Geometry field names associated with a table may be listed to a file:

```
x.list_geoms(a_partition_name, a_table_name, a_filename)
```

To set the output directory for dumps:

```
x.set_output_dir(a_path)
```

To dump a single table:

```
x.dump_table(a_partition_name, a_table_name, _optional
    a_filename, a_restrict_option)
```

where A_RESTRICT_OPTION is either _false (default), _true, or a geometry name. If the option is _false, the entire table is dumped. If it is a geometry name, the table is restricted by that geometry field to either the bounds or the view or to the area specified by method set_extents() (see below). If it is _true, a default geometry field is identified as in the *Standard Dump*.

To set the restriction boundary to a region other than the view extents:

```
x.set_extents(xmin, ymin, xmax, ymax)
```

To clear the extents:

```
x.clear_extents()
```

The *Standard Dump* is invoked as follows:

```
x.standard_dump()
```

and the *Complete Dump*:

```
x.complete_dump()
```

Dump processes may be interrupted using the interrupt_engine() method, and the status may be monitored using the .status_string property. Reporting options may be set as follows:

```
x.set_report_option(_optional option, log_file)
```

OPTION may be one of the following values:

```
:cmd—status and errors are reported to the command window (default)
:cmdfile—status and errors are reported to the command window and errors are
  additionally written to LOG_FILE
[the :gui and :guifile options are for internal use only]
```

Example B.1. Visual Basic 6.0 Application Using the CLI Commands

```
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

Private Sub Command1_Click()

    ' Smallworld Automation must be installed as described in
    ' the Smallworld Automation documentation.
    ' An open Smallworld session must be active and the TICS
    ' server started using ole_admin.start().

    ' Set up Variables

    Dim App As SmallApplication
    Dim Magik As SmallConnection
    Dim MagikRep As SmallObject
```

```
    Dim XMLGenerator As SmallObject
    Dim x As Object
    Dim Temp As String

    ' Establish Connection

    Set App = New SmallApplication
    Set Magik = App.Connect

    ' Load XMLGenerator

    Set XMLGenerator = Magik.GetMagikObject("xmlgenerator")
    If XMLGenerator Is Nothing Then
        Set MagikRep = Magik.GetMagikObject("magik_rep")
        MagikRep.load_file ("d:\0home\test\ps\xmlgenerator.magik")
        Set XMLGenerator = Magik.GetMagikObject("xmlgenerator")
    End If
    Set x = XMLGenerator.new_for_session

    ' **** Example of activating the GUI

    ' x.activate

    ' **** Examples of using CLI methods

    ' List tables and geoms for a table

    x.list_tables "d:\0home\temp\tables.txt"
    x.list_geoms "arizona_gas", "parcel", "d:\0home\temp\geoms.txt"

    ' Export table using given extents

    x.set_output_dir "d:\0home\temp"
    x.set_extents 43890994#, 389911810#, 43895029#, 389914420#
    x.dump_table "arizona_gas", "parcel", Nothing, "extent"
    Wait_For_It x

    ' Standard dump using view extent

    x.clear_extents
    x.standard_dump
    Wait_For_It x

    ' Complete dump with error logfile

    x.set_report_option ":cmdfile", "errors.txt"
    x.complete_dump

    ' Wait a few seconds, get the status_string, and interrupt

    Sleep 10000
    Temp = x.status_string  'Should be something other than "Done."
    x.interrupt_engine
    Temp = x.status_string  'Should be "Done."

    ' Disconnect

    Magik.Disconnect
    Set Magik = Nothing

End Sub


Private Sub Wait_For_It(ByVal x As Object)
```

```
   Dim IsDone
   Do
       Sleep 2000
       IsDone = (x.status_string = "Done.")
   Loop While Not IsDone

End Sub
```

Index

A

Add, 6, 9
Add Item, 9
Annotating Sketches, 60
AnnotationSymbols.tof, 60
Apply, 6, 23
Apply Changes, 7
AutoCAD, 12
  Block Explosions, 36
AutoCAD Colors, 75
AutoCAD Drawings
  Drawing Extents
    Calculations, 51
AutoLSP, 64

B

Back Ups, 49
Background Color, 18
Bird's-eye View, 8, 20
Block Explosions, 36
Browse Box, 9
Button
  Add, 6, 9
  Add Item, 9
  Apply, 6, 7, 23
  Clear, 6
  Clear All, 6
  Delete, 6, 7, 9
  Exclude, 9
  Exit, 6, 7, 9, 9, 23
  Find, 6
  Help, 6
  Hide, 6
  Include, 9
  Remove, 9
  Remove Item, 9
  Save, 6, 7, 23
  Save Configuration, 9
  Show, 6
Buttons
  Elevator, 6, 8
  Min/Max, 6
  Mode, 6
  Pan, 6, 8

C

Centering Point, 18
Clear, 6
Clear All, 6
ClientLogo.png, 61
Color, 23
Colors, 18, 75
Comma-Separated File (CSV), 62
Company Logo, 61
Configuration Windows, 9
Configure Map Viewing, 19
Configure Translator, 19
Control Panel, 3, 8
Coordinates, 8
CSV data source, 38
CSV File, 62, 62
Customer Information Systems, 37

D

Data Properties, 29
Data Source, 4
  Merge Transform, 40
Data Type, 4
Data Types, 27
  Splitting, 51
DBF data source, 38
Delete, 6, 9
Delete Items, 7

E

Elevator Buttons, 6, 8
Email Notification, 48
ESRI
  Ignoring Tables, 56
  Rotation Fields, 56
Excel data source, 38
Exclude, 9
Exit, 6, 9, 23
External Data
  Merging, 37
  Merging Problems, 43

F

Field, 4
File Sets, 12
Files
  AnnotationSymbols.tof, 60
  ClientLogo.png, 61
  find-item.png, 61
  MapControl.tof, 51
  Nomenclature.tof, 51
Find, 6
Find Item Arrow, 61
Find Items, 35, 37
  Creating New, 35
  Removing, 36
  Selecting Sources, 36
  Setting Conditions, 35
find-item.png, 61

G

Geodatabase, 12
Grid Cells
  Properties
    columns, 51

GridLevel, 51
lowerLeft, 51
multiplier, 51
nameMapping, 51
numeric, 51
prefix, 51
rows, 51
start, 51

H

Help, 6, 9
Hide, 6
Hover Indicator, 8

I

Icons, 23
Include, 9
Installing Map Viewer, 45
Inventory Databases, 37

J

Jobs Data, 59

K

Key Map, 22

L

Label, 4
Labelling, 29
Adding to Translation, 33
Creating New Labels, 29
Examples, 33
Justification, 32
Location, 30
Naming, 30
Placement, 30
Spacing, 32
Text Settings, 31
Labels, 29
Legend Items, 21
Level of Detail, 22
Lines, 23
LOD Contents, 20
Logo, 20
Company, 61

M

Maintenance, 45, 49
Map
Properties
height, 51
originX, 51
originY, 51
width, 51
Map Navigation
Buttons
Elevator, 6
Elevator Buttons, 8
Min/Max, 6
Mode, 6
Pan, 6
Coordinates, 8
Deflection, 8
Hover Indicator, 8
Map Grid Numbers, 8
Zoom Level, 8
Map Translator Control Panel, 10, 15
Map Viewer, 8
Installing, 45
Installing from CD, 45
Installing from Network, 46
Running from Network, 46
Updating, 45, 46
Map Viewer Labels, 29
Map Viewing Configuration, 23
Mapping Features, 60
Mapping Symbols
Storage Location, 61
MapTranslator.tof, 55
Min/Max Buttons, 6
Mode Buttons, 6

N

Name, 3
Navigation, 22
Nomenclatures
Grid Based , 51

O

ODBC Data Source, 37
Oracle
Merging into Translator, 14

P

Pan Buttons, 6, 8
Picklist, 9
Pictures
Adding to Map Viewer, 57
Points, 23
Post Maps to Network, 45
Propagation, 43
Property Naming, 29
Python Scripts, 68
Error Handling, 71
Example
Building a Transform, 74
Building Another Transform, 41
Counting Number of Objects, 74
Multiple Merge Fields, 73
Regular Expression, 73
Splitting Graphical Elements, 72
Telephone Number, 72
String Manipulations, 70

R

Raster Images
  Adding to Map Viewer, 57
Reconfiguration, 45
Redline Symbols, 60
  Storage Location, 61
Remove, 9
Remove Item, 9
Removing Find Items, 36
Reopen All Drawings, 49
Requirements
  System, 3
  User, 3

S

Save, 6, 23
Save Configuration, 9
Saving Changes, 8
Scale, 20
Scaling, 19
Scheduled Translation, 46
SDE Database, 11
ShapeFileImport.lsp, 64
Shapefiles, 13, 62, 64
Show, 6
Smallworld, 76
Sources
  Non-GIS, 37
Special Characteristics, 60
Specify Data Sources, 10
Splitting
  Data Types, 51
  Styles, 51
SQL data source, 38
Style, 4
Styles
  Splitting, 51
Symbols
  Annotating Sketches, 60
  Creating
    Construction Symbols, 60
    Redline Symbols, 60
  Mapping
    Storage Location, 61
  Presenting Mapping Features, 60
  Redline
    Storage Location, 61

T

Tab
  Primary, 5
  Secondary, 5
Text, 23
Textbox, 9
TOF Files, 51
Transforms
  Building with Python, 41
Translate Maps, 15, 45
Troubleshooting, 49

U

Updating Map Viewer, 45

V

View Data, 27
View Maps, 45
Visibility, 19

X

XML data source, 38
XMLGenerator, 76

Z

Zip Up Configuration, 49
Zip Up Logs, 49
Zoom Level, 20
Zooms, 23